Strain Distribution Around Underground Openings

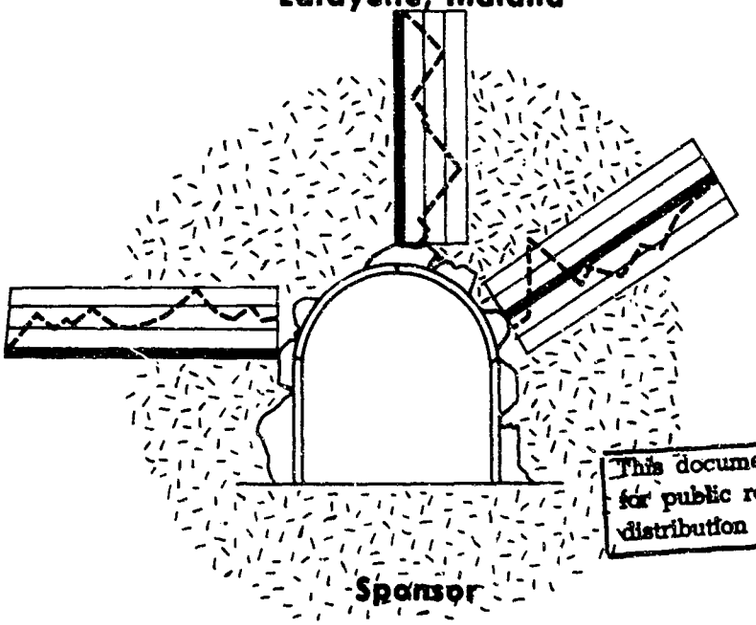
Technical Report No. 4

STATISTICAL METHODS TO COMPILE AND CORRELATE ROCK PROPERTIES

—COMPUTER TECHNIQUES—

Patricia Nahas

Soil and Rock Mechanics Area
School of Civil Engineering
Purdue University
Lafayette, Indiana



Sponsor

ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE

May 1970

Prepared For
OFFICE OF THE CHIEF OF ENGINEERS
DEPARTMENT OF THE ARMY
WASHINGTON, D.C.

Strain Distribution Around Underground Openings

Technical Report No. 4

STATISTICAL METHODS TO COMPILE AND CORRELATE ROCK PROPERTIES

- COMPUTER TECHNIQUES -

Patricia Nahas

Soil & Rock Mechanics Area
School of Civil Engineering
Purdue University
Lafayette, Indiana

Contract No. DACA 73-68-C-0002(P002)

Sponsor

ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE

May 1970

Prepared For

OFFICE OF THE CHIEF OF ENGINEERS
DEPARTMENT OF THE ARMY
WASHINGTON, D.C.

TABLE OF CONTENTS

	Page
Abstract	
1. Introduction	1
2. Data Storage	1
3. Descriptive Programs	
3.1 Listing of Tape	2
3.2 Count Programs	3
3.3 Ranges and Mean Values	5
4. Computational Programs	
4.1 Least Squares	7
4.2 Histograms and Scattergrams	8
5. Comments and Recommendations	10
Appendix	1A

Abstract

A data tape is necessary for the storage of a systematized collection of physico-mechanical properties of rocks. Specific programs permit the obtaining of descriptive information on the data - ranges, means, and counts. Statistical routines yield histograms, scattergrams, and least squares equations. One objective is to provide information that can form the basis for some degree of uniformity in such research. Experience with the programs yielded certain principles and changes that would improve efficiency. It is recommended that one choose an efficient means of data storage, maintain a back-up data source and precise records, run all descriptive programs first, process as many cases per run as possible, determine common scales where necessary for later interpretation, and label output meaningfully.

1. INTRODUCTION

Several requests have been made for information on how the statistical analyses used in Technical Report No. 2 have been handled, especially with reference to the computer techniques involved. The information in this supplement to Technical Report No. 2 is a brief summary of these techniques. It includes both specific examples of computer programs used to obtain the results presented in Technical Report No. 2 and commentary on variations and possible improvements. These programs are intended as a useful guideline for those doing similar research on their own data at their own facility.

2. DATA STORAGE

The preliminary work involved transferring the data to magnetic tape for efficient storage. First, the data were recorded on the coded sheets described in Appendix A of Technical Report No. 2. The data on these sheets was then keypunched on standard 80-column IBM cards. A scratch (temporary) tape was then created from these cards. The data on this tape was subjected to the standard UTILITY SORT routine for the IBM 7094. This procedure provided a second scratch tape which contained the information sorted according to rock type. The final operation involved the use of a COBOL program to create a magnetic tape containing the original data, sorted on rock type, in such a manner as was desired for print-out. Note that the SORT routines often distinguish between a blank and a zero. Hence, it is to one's advantage to be consistent in this respect when recording the data on the coded sheets. For example, in our case, a three-character field was allowed to accommodate the number identifying different rock types. When this number contained only two digits, say 22, it was sometimes entered as 022 and sometimes as _22. Either way

provide the necessary identification for the corresponding data. But since the SORT routine considers these as two different things, the group of data for rock type #22 on our tape is split, an undesirable situation as part appears in one section of the tape and part in another.

Several copies of this tape have since been made because constant use eventually results in bad spots that produce READ PARITY errors. (One good copy should always be reserved as a back-up tape.) Also, it was convenient to have several tapes available in order to speed up the computer turn-around time. For example, if each of two programs requested either tape #544 or tape #630, it was possible to have both programs running at once, each using the same information but from different tapes mounted on different tape drives. This could not have been done if both programs requested tape #544 only. One of the programs would have had to await the return of tape #544 before being run. (Our original tape was created for the IBM 7094 because that was the computer in use at that time in our Computer Sciences Center. Since then, the CDC 6500 has replaced the 7094 as the principal computer being used. We are now in the process of converting our tape to one internal to the CDC in hopes that computer usage of the tape will be facilitated. Then, because of better turn-around time, it may no longer be necessary to maintain several tapes.)

3. DESCRIPTIVE PROGRAMS

3.1 LISTING OF TAPE (pp. 13-15)

The first step after creating the data tape is to obtain a complete listing of the tape. Only in this way will you be assured that all the data have been successfully transferred to the tape. Such a listing will also provide a useful record for future reference. It should be noted again that one must always maintain a duplicate tape as a back-up tape. Accidents

or repeated usage can result in a tape that can no longer be successfully read by the computer.

After having obtained a listing of the tape, one may want to run programs that will provide for detailed information about the data on the tape. Several of the included programs are of this nature, while others perform certain computations on the data. Each program will now be considered individually. In the Conclusion, I will comment on other similar programs that could be run and on certain approaches or changes that in retrospect would have been more efficient and certainly less time consuming.

3.2 COUNT PROGRAMS

Program # 1 (pp. 16-23) is run to obtain information on the number of non-blank entries present for each of the variables on the tape. Each variable on each of the eight data cards per set[1/] is read and compared with a constant, B, that is preset to blank. If the variable in question is non-blank, i.e., not equal to B, a counter is increased. (Note that the counters are initialized to zero as a precaution.) Otherwise, the counter maintains its value and the program proceeds to the next test, until all variables have been compared. The program then reads another record from the tape and proceeds as above. This process is repeated until all the data on the tape is exhausted. The output from the program consists of a listing of the variable names and the associated counts or N's. (Note that the variables are tested against a blank value and not zero because zero is a legitimate entry, i.e., a blank indicates no data whereas zero is an acceptable value.)

If such a count is needed only for certain selected variables, then one need read only those particular values from

---

1/ Note that the eight cards that are referred to as a "data set" in Technical Report No. 2 will be referred to as a "record" with respect to the data tape.

the tape. (See program 1a, pp.24-26). Depending upon the manner in which your data is set up on your tape, it may or may not be necessary to provide for reading dummy information in order to skip to the next record. For example, suppose we wish to concern ourselves only with a variable - say A20 - on the A card. Then in order for our DO LOOP to execute properly, one of two alternatives is necessary. We could have the following READ and FORMAT statements:

```
      READ (1,1) A20
1     FORMAT (10X,A4///////)
```

where the seven slashes would cause the computer to skip the next seven cards and begin reading again with the first card of the next record, or we could read dummy variables as follows:

```
      READ (1,1) A20
      READ (1,3) B
      READ (1,3) C
      READ (1,3) D
      READ (1,3) E
      READ (1,3) F
      READ (1,3) G
      READ (1,3) H
1     FORMAT (10X,A4)
3     FORMAT (50X,A5)
```

where the tape unit we are reading from is Tape 1. The latter alternative is selected as safer since the interpretation of the slashes is subject to variation depending on the particular computer.

Program #2 (pp. 27-30) is a slight variation of Program #1. For certain chosen variables, it is necessary to know how many cases there are of non-blank entries for both variables for the same data set. For this program, the required variables for each record are read and then each variable for the particular pair is compared against the blank. If both are non-blank, the

counter is incremented. Otherwise, the program goes on to the next test. After all pairs for a data set are processed, the variables for the next data set are read and compared. This procedure is repeated until all data on the tape is used. The output is a listing of the values of N for the various pairs. Note that here, as previously, the reading and comparisons are performed within the same DO LOOP. Because of the large quantity of data, it is not feasible or even possible to read the data into arrays and then perform the comparisons separately.

One further variation of these programs may be useful. It may be necessary to obtain such information for the variables within a particular rock type rather than over all rock types. Such can be accomplished by reading in the rock type variable along with the data in question and testing its value against the value identifying the rock type of interest. The sequence of instructions would be as follows:

```
      DO 100 K = 1,2170
      READ (1,1) IRT,A20,A1
      READ (1,3) BB
       .
        .
         .
      READ (1,3) HH
1     FORMAT (15X, I3, 10X, A4, A3)
3     FORMAT (50X,A5)
      IF (IRT .NE. 223) GO TO 100
      IF {(A20 .NE. B) .AND. (A1 .NE. B)} N = N + 1
100   CONTINUE
```

where IRT is the variable for rock type and 223 is the value of the rock type of interest.

3.3 RANGES AND MEAN VALUES

Program #3 (pp. 31-35) and Program #4 (pp. 36-43) provide further information about the data on the tape. Program

#3 is designed to print out all values within specified ranges for a particular selection of variables and the associated rock types. In addition, the program computes for each variable the maximum and minimum values, the number of values, and the mean over these values.

At this point, it is important to understand that the data on our tape was entered in such a manner that it could be read off only in integer or A field format. The variables could not be read off as real variables. For all homogeneous computations (i.e., computations involving only one variable) it was not necessary to perform any scaling in order to obtain meaningful results. Any scaling could be applied directly to the results. So if the entry 432 for variable B8 really represented 4.32 and we got a mean of 397 for B8, we knew immediately that the actual mean was 3.97. However, if the computation involved more than one variable, it was mandatory that the variables be scaled before the computation began. This fact will be discussed later in conjunction with several of the computational programs. Also discussed in conjunction with the computational programs is the reason for specifying ranges for certain of the variables. (pp. 7, 9 and 10)

Program #4 differs from Program #3 in that the print-out is limited each time to a particular rock type. This permits the comparison, for example, of the ranges of values for a particular variable over the whole tape with the range within a specific rock type. The actual differences in the program are modifications to test for rock type and to reinitialize counters. It is, of course, possible that some variables for a particular rock type would be blank, and such a situation has to be taken into account. It is also possible that the last non-blank record read from the tape would be of a different rock type than the preceding. In that case, it is necessary to provide a way for printing out the last information. By inserting the test on END OF FILE, it is possible to reinitialize all

values, go back and perform the tests, and then print out the desired information. The test on END OF FILE guarantees that this last information will not be lost either through exceeding the DO LOOP limits or through trying to read past the END-OF-FILE on the tape. (Note that the DO LOOP parameter goes from 1 to 2170 previously and from 1 to 2171 in this case to allow for this last possibility.)

The following are also to be noted. The manner in which the DO LOOP parameter is set up requires that the first time through the loop you test IRT(1) against IRT(0). (The IRT array represents the values identifying the different rock types.) To make this test possible, we equivalence the first element of the IRT array, i.e., IRT(1) with the second element of a two-element array called DUMMY. Hence IRT(1) = DUMMY(2). Then, IRT(0) is equivalent to DUMMY(1) which is set equal to the value of the first rock type on the tape. In this way, the first time the program reads the rock type, it reads IRT(1) = 3 and when it goes to compare IRT(1) with IRT(0), the computer has a value for IRT(0) and it is 3. From there on, all proceeds as usual.

4. COMPUTATIONAL PROGRAMS

4.1 LEAST SQUARES

Program #5 (pp. 24-26) also reads selected variables from the tape and compares certain pairs with blank. When both variables in the pair for the same data set are non-blank, the values are stored in arrays. However, the values in this case must be scaled so that the Least Squares Equation will make sense. For example, if 429 for B15 represents the real value of 4.29 and 364 for A1 is actually 36.4, this difference in decimals must be maintained for the resulting equation to be meaningful. In addition, the associated rock types are stored in another array, and this information (.i.e., the pairs of points and the rock types for which they occur) is part of the print-out. The values in these arrays are the input data for the Least Squares

routine. The routine computes the least squares coefficients A and B of the equation Y = AX + B from this data. Note that the comment cards in the Least Squares program deck adequately explain its source and setup. As many equations as desired can be obtained in a particular run, subject of course to the storage limits and time considerations of a particular computer. It should be noted that the N for a particular pair of variables must be known in advance. This information is needed not only for the dimensioning of the arrays but must also be fed as a parameter to the Least Squares routine. Hence, we see the value of the Count Programs.

4.2 HISTOGRAMS AND SCATTERGRAMS

These Count programs are also necessary for the histogram and scattergram routines which follow. These require the number of data points as one of their input parameters. The programs are set up in such a way that the first part of both reads the selected variables from the tape and makes the comparisons with the blank. For those cases where both variables have non-blank entries for the same data set, the program stores the values in arrays. After the whole tape has been read, these arrays are put on the disk and serve then as input for the parts of the program that produce the scattergrams and histograms. These parts of the routines are taken from: "BIOMEDICAL COMPUTER PROGRAMS, HEALTH SCIENCES COMPUTING FACILITY, DEPARTMENT OF PREVENTIVE MEDICINE AND PUBLIC HEALTH SCHOOL OF MEDICINE, UNIVERSITY OF CALIFORNIA, LOS ANGELES, JANUARY 1, 1964", W. J. Dixon, Editor. Included with the print-out of these programs as they have been adapted to run on our computer are the user's write ups. These provide information on how to set up the data cards, and the various options available. For example, the program that gives the scattergram output could be run just to obtain correlation coefficients.

You will also note that these BMD routines are on a library file at our facility and are loaded as part of the user's program via the control card

LIBCOPY(STATBIN,LGO,BMD2D)

Hence, the program deck for these two routines is set up as follows:

Control Cards

7_{8_9}

Program Deck

7_{8_9}

Data Cards for BMD Routine

$6_{7_{8_9}}$

This set up is not immediately obvious from the print-out sheets, especially since the BMD routines are not printed out each time.

Another important point is that these routines accept only real variables as input. That is, the data format card is valid only in Fw.d format. It does not matter that the variables in the first part of the program are written on the disk in A format. The BMD routine will automatically convert them as it reads them. What is necessary though is that the Fw.d format accurately represents the true value of the variables with respect to each other. For example, suppose two variables are written on the tape via the format (A4,A3). Then if the values 1492 and 765 really represent 14.92 and 76.5, they must be read by the format (F4.2,F3.1) and <u>not</u> by (F4.0,F3.0). The former will permit the BMD routine to perform meaningful computations, whereas the latter will not.

> 1492 - 765 = 727 would be an erroneous and meaningless computation because the real computation is
> 14.92 - 76.5 = -61.58

The histogram program is also run to obtain a master histogram for each variable on the tape. (For further explanation refer to Section 4.4 in Technical Report No. 2) The program is

a simple variation of the one included. As before, one needs to know the number of non-blank entries for the variable in question. Then it is a simple matter of reading the tape and transferring the values to the disk for input to the histogram. In several cases in our computations, upper and lower limits were set for the variable and each value was tested not only against the blank but also against the desired range. This testing was necessary to eliminate bad entries. These latter had the unfortunate effect of altering the scaling on the histogram in such a way as to make the graph meaningless. For example, suppose the range of the selected variable was 0 to 20, and that a value of 17.4 had been erroneously entered as 174.0. Then instead of being evenly spread, the histogram would be squeezed together on the left, with just one value to the far right, a situation which is of little help for interpretation and comparison. In addition, the histogram program can be run to obtain histograms for variables within a particular rock type. All that is required is the insertion of a test on rock type similar to the one previously described.

5. COMMENTS AND RECOMMENDATIONS

Experience with the preceding programs has led to suggestions which we plan to use in the continuing research on these rock problems. We expect these suggestions to improve the efficiency of the operation and to prevent the reader from repeating some of our mistakes.

RECOMMENDATION NO. 1

All programs giving descriptive information about the data on the tape should be run first. In this way, erroneous or extreme entries will be discovered immediately and the tape can be corrected or the subsequent programs can be written with tests that will screen out such values. This will eliminate the need to rerun programs processed before these errors are discovered.

RECOMMENDATION NO. 2

Investigate the possibility of creating a permanent file containing one's data. In this way, one eliminates the need to mount a tape each time a program is run, and turn-around time should be correspondingly faster. We plan to try this approach with the next group of programs we will be running.

RECOMMENDATION NO. 3

Process more than one histogram or scattergram per run by performing comparisons on several pairs of variables at once and writing the arrays on the disk in order, one after the other. Then include in your deck data cards for each of the successive histograms or scattergrams. (Note the repeat specifications in the write-ups accompanying the BMD routines.) This technique has been tested successfully in connection with another problem.

RECOMMENDATION NO. 4

Keep a back-up tape.

RECOMMENDATION NO. 5

Keep careful and precise records. These may take in part the form of comment cards in programs. But they may also be written notations, etc. Only in this way is one able to go back after a period of time and recall accurately what he did[2/]. And in addition, only in this way can the information be successfully handed over to another person.

RECOMMENDATION NO. 6

Determine a common scale for each of the base variables in the Comparison Histograms. Use the scale obtained in the Master Histogram for the variable whenever that variable

---

[2/] This is especially important in our situation, because Purdue's computer center operation is in part run for student learning in addition to user service. Thus, there is constant change in the system and in available routines. This means that slight changes may be necessary any time a program is rerun after a time lag.

appears in a Comparison Histogram. Note that the SELECT card we use specifies an impossibly small interval so that the BMD program will supply optimum scaling for the values in question. This approach was necessary since we had no advance idea as to the ranges of the variables once the comparisons for the pair had been made. In retrospect, we realize that we could have expanded the program that made the original counts and had it give us maxima and minima also.

RECOMMENDATION NO. 7

Use two label cards for the Comparison Histograms. The first will identify the variable in the particular histogram. The other will indicate the variable of comparison, i.e., the variable against whose values the histogram variable was compared.

COMMENTS

A few additional comments are directed toward the BMD routines. It would be convenient if we could feed these a variable N for the number of data points. If this were possible, we could eliminate the Count programs and simplify the first part of the present scattergram and histogram routines. This task is one to be tackled in the future. It may, however, prove more advantageous to replace the BMD routines with ones of our own, possibly utilizing the Calcomp Plotter for the graphing.

It is hoped that the above will be helpful to all who are involved in similar research. Perhaps some degree of consistency and uniformity can be achieved which will make the research understandable to all involved and which will permit comparisons. It is also hoped that useless and time-consuming duplications of effort will thus be avoided. Much can still be done in the area of programming techniques. What is included here is by no means exhaustive or definitive. But it is set forth with the hope that it is a start toward communication and a help toward further research.

PARTIAL LISTING OF DATA TAPE

```
0 1314
M 1314
A 1424 B 17   1003  0  0
B 1424
C 1424
D 14241013        1    97                          1       100+11
E 1424
F 1424
G 1424
H 1424
A  849 U  1 511004635  0109  266
B  849
C  849
D  8491012    3242                                1      1196+26
E  849
F  849
G  849
H  849
A 1303 K  6  91005  0  02    1       261   250          41    17   71
B 1303
C 1303
D 13031013    31  1   208
E 1303
F 1303
G 1303
H 1303
A  152 A  1 102007  0  0208  1  1          297          04          75  29   79
B  1521562  9032112         176    3   124  491
C  152
D  1521012  9032521   389
E  152
F  152
G  152
H  1521102  903252                                                  2   66
A  130 A  1  91007  0  0108054  13          285          09          69  15  13
B  1301562  9032111         152    4    892 389
C  130
D  1301012  9032521   334
E  130
F  130
G  130
H  1301102  903252                                                  2   38
A   92 B  9 121007  0  0101052 51                   254  137    57
B   921552    33111                  +20 342 151
C   92
D   921022    33421    24   03  46 06  59 18 1181   02 200+01
E   92
F   921 04 235+05             08 276+10      106
G   92
H   921112    33421   012
A   91 B  9 121007  0  0101052 51                   254  137    57
B   911552    33111                  +25 340 263
C   91
D   911022    33421    99   03 117 09 157 18 1801   02 439+11
E   91
F   911 04 531+13             08 582+15      103
G   91
H   911112    33421   032
A   57    3  91007  0  0102  1 61                   262   36
B   571552    33111                  +11 378 172
C   57
D   571022    33421    97   06 154 13 220 20 2871   03 437+08 333
```

```
E   57
F   571 05 460*09 435          08 476*10 469          15 514*13 545 111
G   57
H   57
A    4 B  1 301007  0  0108  1 612          274          16
B    4
C    4
D    41022  1033421   324   20 456 40 588 60 7201   10 91 *25
E    4
F    41 20 92 *25              30 91 *24              60 87 *22
G    4
H    4
A    5 B  1 301007  0  0108092 512          272          45
B    5
C    5
D    51022  1033421   246   20 394 40 542 60 6901   10 91 *25
E    5
F    51 20 92 *25              30 91 *24              60 87 *22
G    5
H    5
A   58 B  3  91007  0  0102  1 51                    256   61
B   581552    33111                    *11 378 172
C   58
D   581022    33521    54   06 115 13 186 20 2581   03 437*08
E   58
F   581 05 460*09              08 476*10              15 514*13
G   58
H   58
A   59 B  3  91007  0  0102  5 22                    206  248
B   59
C   59
D   591012    33521    33                          1   03 030*01
E   59
F   591 05 031*01              08 033*01              15 038*01
G   59
H   59
A  153 A  1 102007  0  0208  1              291                          84       12
B  1531562  9032112          161    4   102  429
C  153
D  1531012  9032521   520
E  153
F  153
G  153
H  153
A  154 A  1 102007  0  0208  1                                           77       73
B  154
C  154
D  1541012  9032521   374
E  154
F  154
G  154
H  154
A  155 A  1 102007  0  0208055                                         53
B  155
C  155
D  1551012  9032522   118
E  155
F  155
G  155
H  155
A  455 A  4  71007  0  0102  A 5            286                          71       14
B  455
```

```
C  455
D  4551012  9032522   282                          2      1005+18
E  455
F  4551141 994*20             282 977*19
G  455
H  455
A  259 A  2  92007614  0205055                270          50          49    9   36
B  2591562  9032112          127   20*09 59  268
C  259
D  2591012  9032521   173                          2      67 +16
E  259                                                     1
F  2591 87 68 *20             173 62 *26
G  259
H  2591102  903252                                                          2   21
A  260 A  2 102007608  0205055                280          04          45  25   88
B  260
C  260
D  2601012  9032521   496                          2      87 +15
E  260
F  2601248 83 *16             496 80
G  260
H  2601102  903252                                                          2   41
A  258 A  2  92007674  0205055                204          180              3   19
B  2581562  9032112          58     -04 091 048
C  258
D  2581012  9032522    25                          2      025*26
E  258
F  2581 13 028*34              25 028*42
G  258
H  2581102  903252                                                          2   03
A  993 P  1 172007  0  02  05
B  993
C  993
D  993
E  993
F  993
G  993
H  9931112      521   070
A  817 D  29762007  0  02       2                          75
B  817
C  817
D  817
E  817
F  817
G  817
H  817
A  818 D  29762007  0  02       3          264          81
B  818
C  818
D  818
E  818
F  818
G  818
H  818
A  816 D  29682007634  02       3          294          34          71        14
B  816
C  816
D  8161012    32421   282                          1   09    +181005
E  816
F  8161 26    *17 977         43    *20 994          77    *19 977
G  816
H  816
```

Program No. 1

COUNT PROGRAM

Number of non-blank entries for each variable on a tape

```
                PROGRAM VC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
000003          DIMENSION N(78)
000003          DATA B/1H /
000003          DO 10 K=1,78
000005       10 N(K) = 0
000010          DO 100 K=1,2170
000012          READ(1,1) A1,A2,A3,A4,A5,A6,A7,A8,A9
000037        1 FORMAT(32X,A1,3X,A4,2X,A4,2X,A4,2X,A3,2X,A4,2X,A3,2X,A2,2X,A3)
000037          READ(1,2) B1,B2,B3,B4,B5,B6,B7,B8,B9
000065        2 FORMAT(28X,A3,A3,A2,A3,A4,A4,14X,A3,2X,A3,2X,A3)
000065          READ(1,2) C1,C2,C3,C4,C5,C6,C7,C8,C9
000113          READ(1,3) D1,D2,D3,D4,D5,D6,D7,D8
000137        3 FORMAT(21X,A4,5X,A4,3X,A4,3X,A4,6X,A4,A3,A4,2X,A2)
000137          READ(1,3) E1,E2,E3,E4,E5,E6,E7,E8
000163          READ(1,4) F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12
000217        4 FORMAT(10X,A4,A3,A4,2X,A2,7X,A4,A3,A4,2X,A2,7X,A4,A3,A4,2X,A2)
000217          READ(1,4)G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12
000253          READ(1,5) H1,H2,H3,H4,H5,H6,H7,H8,H9,H10,H11
000305        5 FORMAT(21X,A4,6X,A4,A3,A4,3X,A4,A3,A4,3X,A4,A3,A4,3X,A3)
000305          IF(A1 .NE.B) N(1 ) = N(1 ) + 1
000311          IF(A2 .NE.B) N(2 ) = N(2 ) + 1
000315          IF(A3 .NE.B) N(3 ) = N(3 ) + 1
000321          IF(A4 .NE.B) N(4 ) = N(4 ) + 1
000325          IF(A5 .NE.B) N(5 ) = N(5 ) + 1
000331          IF(A6 .NE.B) N(6 ) = N(6 ) + 1
000335          IF(A7 .NE.B) N(7 ) = N(7 ) + 1
000341          IF(A8 .NE.B) N(8 ) = N(8 ) + 1
000345          IF(A9 .NE.B) N(9 ) = N(9 ) + 1
000351          IF(B1 .NE.B) N(10) = N(10) + 1
000355          IF(B2 .NE.B) N(11) = N(11) + 1
000361          IF(B3 .NE.B) N(12) = N(12) + 1
000365          IF(B4 .NE.B) N(13) = N(13) + 1
000371          IF(B5 .NE.B) N(14) = N(14) + 1
000375          IF(B6 .NE.B) N(15) = N(15) + 1
000401          IF(B7 .NE.B) N(16) = N(16) + 1
000405          IF(B8 .NE.B) N(17) = N(17) + 1
000411          IF(B9 .NE.B) N(18) = N(18) + 1
000415          IF(C1 .NE.B) N(19) = N(19) + 1
000421          IF(C2 .NE.B) N(20) = N(20) + 1
000425          IF(C3 .NE.B) N(21) = N(21) + 1
000431          IF(C4 .NE.B) N(22) = N(22) + 1
000435          IF(C5 .NE.B) N(23) = N(23) + 1
000441          IF(C6 .NE.B) N(24) = N(24) + 1
000445          IF(C7 .NE.B) N(25) = N(25) + 1
000451          IF(C8 .NE.B) N(26) = N(26) + 1
000455          IF(C9 .NE.B) N(27) = N(27) + 1
000461          IF(D1 .NE.B) N(28) = N(28) + 1
000465          IF(D2 .NE.B) N(29) = N(29) + 1
000471          IF(D3 .NE.B) N(30) = N(30) + 1
000475          IF(D4 .NE.B) N(31) = N(31) + 1
000501          IF(D5 .NE.B) N(32) = N(32) + 1
000505          IF(D6 .NE.B) N(33) = N(33) + 1
000511          IF(D7 .NE.B) N(34) = N(34) + 1
000515          IF(D8 .NE.B) N(35) = N(35) + 1
000521          IF(E1 .NE.B) N(36) = N(36) + 1
000525          IF(E2 .NE.B) N(37) = N(37) + 1
000531          IF(E3 .NE.B) N(38) = N(38) + 1
000535          IF(E4 .NE.B) N(39) = N(39) + 1
000541          IF(E5 .NE.B) N(40) = N(40) + 1
000545          IF(E6 .NE.B) N(41) = N(41) + 1
000551          IF(E7 .NE.B) N(42) = N(42) + 1
000555          IF(E8 .NE.B) N(43) = N(43) + 1
000561          IF(F1 .NE.B) N(44) = N(44) + 1
000565          IF(F2 .NE.B) N(45) = N(45) + 1
000571          IF(F3 .NE.B) N(46) = N(46) + 1
000575          IF(F4 .NE.B) N(47) = N(47) + 1
000601          IF(F5 .NE.B) N(48) = N(48) + 1
000605          IF(F6 .NE.B) N(49) = N(49) + 1
000611          IF(F7 .NE.B) N(50) = N(50) + 1
000615          IF(F8 .NE.B) N(51) = N(51) + 1
000621          IF(F9 .NE.B) N(52) = N(52) + 1
000625          IF(F10.NE.B) N(53) = N(53) + 1
000631          IF(F11.NE.B) N(54) = N(54) + 1
000635          IF(F12.NE.B) N(55) = N(55) + 1
000641          IF(G1 .NE.B) N(56) = N(56) + 1
000645          IF(G2 .NE.B) N(57) = N(57) + 1
000651          IF(G3 .NE.B) N(58) = N(58) + 1
000655          IF(G4 .NE.B) N(59) = N(59) + 1
000661          IF(G5 .NE.B) N(60) = N(60) + 1
000665          IF(G6 .NE.B) N(61) = N(61) + 1
000671          IF(G7 .NE.B) N(62) = N(62) + 1
000675          IF(G8 .NE.B) N(63) = N(63) + 1
000701          IF(G9 .NE.B) N(64) = N(64) + 1
000705          IF(G10.NE.B) N(65) = N(65) + 1
000711          IF(G11.NE.B) N(66) = N(66) + 1
000715          IF(G12.NE.B) N(67) = N(67) + 1
000721          IF(H1 .NE.B) N(68) = N(68) + 1
000725          IF(H2 .NE.B) N(69) = N(69) + 1
000731          IF(H3 .NE.B) N(70) = N(70) + 1
000735          IF(H4 .NE.B) N(71) = N(71) + 1
000741          IF(H5 .NE.B) N(72) = N(72) + 1
000745          IF(H6 .NE.B) N(73) = N(73) + 1
000751          IF(H7 .NE.B) N(74) = N(74) + 1
000755          IF(H8 .NE.B) N(75) = N(75) + 1
000761          IF(H9 .NE.B) N(76) = N(76) + 1
000765          IF(H10.NE.B) N(77) = N(77) + 1
000771          IF(H11.NE.B) N(78) = N(78) + 1
000775       100 CONTINUE
```

```
000777          WRITE(6,900)
001003      900 FORMAT(1H1)
001003          WRITE(6,1000)
001007     1000 FORMAT(1H ,10A,13HVARIABLE NAME,42X,1HN)
001007          WRITE(6,901) N(1)
001015      901 FORMAT(1H0,10A,12HPERMEABILITY,48X,I4)
001015          WRITE(6,902) N(2)
001023      902 FORMAT(1H0,10A,21HTRUE SPECIFIC GRAVITY,30X,I4)
001023          WRITE(6,903) N(3)
001031      903 FORMAT(1H0,10A,25HAPPARENT SPECIFIC GRAVITY,35X,I4)
001031          WRITE(6,904) N(4)
001037      904 FORMAT(1H0,10A,34HUNSPECIFIED TYPE OF SPECIFIC GRAVITY,24X,I4)
001037          WRITE(6,905) N(5)
001045      905 FORMAT(1H0,10A,8HPOROSITY,52X,I4)
001045          WRITE(6,906) N(6)
001053      906 FORMAT(1H0,10A,10HABSORPTION,50X,I4)
001053          WRITE(6,907) N(7)
001061      907 FORMAT(1H0,10A,21HSCLEROSCOPE HARDNESS,39X,I4)
001061          WRITE(6,908) N(8)
001067      908 FORMAT(1H0,10A,17HABRASIVE HARDNESS,43X,I4)
001067          WRITE(6,909) N(9)
001075      909 FORMAT(1H0,10A,16HIMPACT TOUGHNESS,44X,I4)
001075          WRITE(6,910) N(10)
001103      910 FORMAT(1H0,10A,30HPROPAGATION VEL-LONG WAVES/LAB,30X,I4)
001103          WRITE(6,911) N(11)
001111      911 FORMAT(1H0,10A,36HPROPAGATION VEL-TRANSVERSE WAVES/LAB,24X,I4)
001111          WRITE(6,912) N(12)
001117      912 FORMAT(1H0,10A,29HSPECIFIC DAMPING CAPACITY/LAB,31X,I4)
001117          WRITE(6,913) N(13)
001125      913 FORMAT(1H0,10A,28HPOISSON S RATIO- DYNAMIC/LAB,32X,I4)
001125          WRITE(6,914) N(14)
001133      914 FORMAT(1H0,10A,28HYOUNG S MODULUS, DYNAMIC/LAB,32X,I4)
001133          WRITE(6,915) N(15)
001141      915 FORMAT(1H0,10A,47HMODULUS OF RIGIDITY, SHEAR MODULUS, DYNAMIC/LAB,
                113X,I4)
001141          WRITE(6,916) N(16)
001147      916 FORMAT(1H0,10A,21HSHEAR STRENGTH/STATIC,39X,I4)
001147          WRITE(6,917) N(17)
001155      917 FORMAT(1H0,10A,28HSHEAR STRENGTH AT SS1/STATIC,32X,I4)
001155          WRITE(6,918) N(18)
001163      918 FORMAT(1H0,10A,28HSHEAR STRENGTH AT SS2/STATIC,32X,I4)
001163          WRITE(6,919) N(19)
001171      919 FORMAT(1H0,10A,32HPROPAGATION VEL-LONG WAVES/FIELD,28X,I4)
001171          WRITE(6,920) N(20)
001177      920 FORMAT(1H0,10A,36HPROPAGATION VEL-TRANSVERSE WAVES/FIELD,24X,I4)
001177          WRITE(6,921) N(21)
001205      921 FORMAT(1H0,10A,31HSPECIFIC DAMPING CAPACITY/FIELD,29X,I4)
001205          WRITE(6,922) N(22)
001213      922 FORMAT(1H0,10A,30HPOISSON S RATIO, DYNAMIC/FIELD,30X,I4)
001213          WRITE(6,923) N(23)
001221      923 FORMAT(1H0,10A,30HYOUNG S MODULUS, DYNAMIC/FIELD,30X,I4)
001221          WRITE(6,924) N(24)
001227      924 FORMAT(1H0,10A,49HMODULUS OF RIGIDITY, SHEAR MODULUS, DYNAMIC/FIEL
                1D,11X,I4)
001227          WRITE(6,925) N(25)
001235      925 FORMAT(1H0,10A,27HSHEAR STRENGTH/STATIC/FIELD,33X,I4)
001235          WRITE(6,926) N(26)
001243      926 FORMAT(1H0,10A,34HSHEAR STRENGTH AT SS1/STATIC/FIELD,26X,I4)
001243          WRITE(6,927) N(27)
001251      927 FORMAT(1H0,10A,34HSHEAR STRENGTH AT SS2/STATIC/FIELD,26X,I4)
001251          WRITE(6,928) N(28)
001257      928 FORMAT(1H0,10A,36HCOMPRESSIVE STRENGTH, UNCONFINED/LAB,24X,I4)
```

```
001257          WRITE(6,929) N(29)
001265      929 FORMAT(1H0,10A,41HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS1/LAB,19X,I4
                1)
001265          WRITE(6,930) N(30)
001273      930 FORMAT(1H0,10A,41HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS2/LAB,19X,I4
                1)
001273          WRITE(6,931) N(31)
001301      931 FORMAT(1H0,10A,41HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS3/LAB,19X,I4
                1)
001301          WRITE(6,932) N(32)
001307      932 FORMAT(1H0,10A,33HMODULUS OF DEFORMATION AT US1/LAB,27X,I4)
001307          WRITE(6,933) N(33)
001315      933 FORMAT(1H0,10A,26HPOISSON S RATIO AT US1/LAB,34X,I4)
001315          WRITE(6,934) N(34)
001323          WRITE(6,935) N(35)
001331      934 FORMAT(1H0,10A,26HYOUNG S MODULUS AT US1/LAB,34X,I4)
001331      935 FORMAT(1H0,10A,24HSET FOR UPPER STRESS/LAB,36X,I4)
001331          WRITE(6,936) N(36)
001337      936 FORMAT(1H0,10A,36HCOMPRESSIVE STRENGTH, UNCONFINED/FIELD,22X,I4)
001337          WRITE(6,937) N(37)
001345      937 FORMAT(1H0,10A,43HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS1/FIELD,17X,
                1I4)
001345          WRITE(6,938) N(38)
001353      938 FORMAT(1H0,10A,43HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS2/FIELD,17X,
                1I4)
001353          WRITE(6,939) N(39)
001361      939 FORMAT(1H0,10A,43HCOMPRESSIVE STRENGTH, TRIAXIAL AT LS3/FIELD,17X,
                1I4)
001361          WRITE(6,940) N(40)
001367      940 FORMAT(1H0,10A,35HMODULUS OF DEFORMATION AT US1/FIELD,25X,I4)
001367          WRITE(6,941) N(41)
001375      941 FORMAT(1H0,10A,28HPOISSON S RATIO AT US1/FIELD,32X,I4)
001375          WRITE(6,942) N(42)
001403      942 FORMAT(1H0,10A,28HYOUNG S MODULUS AT US1/FIELD,32X,I4)
001403          WRITE(6,943) N(43)
001411      943 FORMAT(1H0,10A,26HSET FOR UPPER STRESS/FIELD,34X,I4)
001411          WRITE(6,944) N(44)
001417      944 FORMAT(1H0,10A,29HMODULUS OF DEFORMATION AT US2,31X,I4)
001417          WRITE(6,945) N(45)
001425      945 FORMAT(1H0,10A,22HPOISSON S RATIO AT US2,38X,I4)
001425          WRITE(6,946) N(46)
001433      946 FORMAT(1H0,10A,22HYOUNG S MODULUS AT US2,38X,I4)
001433          WRITE(6,947) N(47)
001441      947 FORMAT(1H0,10A,11HSET FOR US2,49X,I4)
001441          WRITE(6,948) N(48)
001447      948 FORMAT(1H0,10A,29HMODULUS OF DEFORMATION AT US3,31X,I4)
001447          WRITE(6,949) N(49)
001455      949 FORMAT(1H0,10A,22HPOISSON S RATIO AT US3,38X,I4)
001455          WRITE(6,950) N(50)
001463      950 FORMAT(1H0,10A,22HYOUNG S MODULUS AT US3,38X,I4)
001463          WRITE(6,951) N(51)
001471      951 FORMAT(1H0,10A,11HSET FOR US3,49X,I4)
001471          WRITE(6,952) N(52)
001477      952 FORMAT(1H0,10A,29HMODULUS OF DEFORMATION AT US4,31X,I4)
001477          WRITE(6,953) N(53)
001505      953 FORMAT(1H0,10A,22HPOISSON S RATIO AT US4,38X,I4)
001505          WRITE(6,954) N(54)
001513      954 FORMAT(1H0,10A,22HYOUNG S MODULUS AT US4,38X,I4)
001513          WRITE(6,955) N(55)
001521      955 FORMAT(1H0,10A,11HSET FOR US4,49X,I4)
001521          WRITE(6,956) N(56)
001527      956 FORMAT(1H0,10A,35HMODULUS OF DEFORMATION AT US2/FIELD,25X,I4)
```

```
001527          WRITE(6,957) N(57)
001535      957 FORMAT(1H0,10X,28HPOISSON S RATIO AT US2/FIELD,42X,I4)
001535          WRITE(6,958) N(58)
001543      958 FORMAT(1H0,10X,28HYOUNG S MODULUS AT US2/FIELD,32X,I4)
001543          WRITE(6,959) N(59)
001551      959 FORMAT(1H0,10X,17HSET FOR US2/FIELD,43X,I4)
001551          WRITE(6,960) N(60)
001557      960 FORMAT(1H0,10X,35HMODULUS OF DEFORMATION AT US3/FIELD,25X,I4)
001557          WRITE(6,961) N(61)
001565      961 FORMAT(1H0,10X,28HPOISSON S RATIO AT US3/FIELD,32X,I4)
001565          WRITE(6,962) N(62)
001573      962 FORMAT(1H0,10X,28HYOUNG S MODULUS AT US3/FIELD,32X,I4)
001573          WRITE(6,963) N(63)
001601      963 FORMAT(1H0,10X,17HSET FOR US3/FIELD,43X,I4)
001601          WRITE(6,964) N(64)
001607      964 FORMAT(1H0,10X,35HMODULUS OF DEFORMATION AT US4/FIELD,25X,I4)
001607          WRITE(6,965) N(65)
001615      965 FORMAT(1H0,10X,28HPOISSON S RATIO AT US4/FIELD,32X,I4)
001615          WRITE(6,966) N(66)
001623      966 FORMAT(1H0,10X,28HYOUNG S MODULUS AT US4/FIELD,32X,I4)
001623          WRITE(6,967) N(67)
001631      967 FORMAT(1H0,10X,17HSET FOR US4/FIELD,43X,I4)
001631          WRITE(6,968) N(68)
001637      968 FORMAT(1H0,10X,16HTENSILE STRENGTH,44X,I4)
001637          WRITE(6,969) N(69)
001645      969 FORMAT(1H0,10X,41HMODULUS OF (TENSILE) DEFORMABILITY AT US1,19X,I4
                1)
001645          WRITE(6,970) N(70)
001653      970 FORMAT(1H0,10X,32HPOISSON S RATIO AT US1 (TENSILE),28X,I4)
001653          WRITE(6,971) N(71)
001661      971 FORMAT(1H0,10X,32HYOUNG S MODULUS AT US1 (TENSILE),28X,I4)
001661          WRITE(6,972) N(72)
001667      972 FORMAT(1H0,10X,41HMODULUS OF (TENSILE) DEFORMABILITY AT US2,19X,I4
                1)
001667          WRITE(6,973) N(73)
001675      973 FORMAT(1H0,10X,32HPOISSON S RATIO AT US2 (TENSILE),28X,I4)
001675          WRITE(6,974) N(74)
001703      974 FORMAT(1H0,10X,32HYOUNG S MODULUS AT US2 (TENSILE),28X,I4)
001703          WRITE(6,975) N(75)
001711      975 FORMAT(1H0,10X,41HMODULUS OF (TENSILE) DEFORMABILITY AT US3,19X,I4
                1)
001711          WRITE(6,976) N(76)
001717      976 FORMAT(1H0,10X,32HPOISSON S RATIO AT US3 (TENSILE),28X,I4)
001717          WRITE(6,977) N(77)
001725      977 FORMAT(1H0,10X,32HYOUNG S MODULUS AT US3 (TENSILE),28X,I4)
001725          WRITE(6,978) N(78)
001733      978 FORMAT(1H0,10X,18HMODULUS OF RUPTURE,42X,I4)
001733          STOP
001735          END
```

PROGRAM LENGTH INCLUDING I/O BUFFERS
006345

UNUSED COMPILER SPACE
012500

LOAD MAP FILE = LGO 13.27.01. 12/19/69. PAGE 1

FWA LOAD 100 LWA LOAD 13452 FWA LOADER 57303 FWA TABLES 56604 UNUSED STORAGE 43132

PROGRAM	ADDRESS	FILE
VC	100	LGO
INPUTC	6445	SYSTEM
SYSTEM	7564	SYSTEM
OUTPTC	10637	SYSTEM
FATAL78	12177	SYSTEM
SIOS	12315	SYSTEM
GETBA	13433	SYSTEM

COMMON	ADDRESS	LENGTH
SCOPE2	7566	
/BLANK/	0	0

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
VC	100								
INPUTC	6447	VC	13	15	17	21	23	25	27	31
			33	35	36	41	43	45	47	51
			53	55	57	61	63	64	67	71
			73	75	77	101	103	105	107	111
			112	115	117	121	123	125	127	131
			133	135	136	141	143	145	147	151
			153	155	157	161	162	165	167	171
			173	175	177	201	203	205	207	211
			213	215	216	221	223	225	227	231
			233	235	237	241	243	245	247	251
			252	255	257	261	263	265	267	271
			273	275	277	301	303	304		
KRAKEN	6551									
QBNTRY	7567	VC	2							
SYSTEM	7774	INPUTC	37	727						
		OUTPTC	14	1144						
SYSTEMC	7740									
SYSTEMP	7767									
END	7663	VC	1736							
STOP	7713	VC	1734							
EXIT	7705									
ABNORML	7723	INPUTC	40	730						
		OUTPTC	15	1145						
OUTPTC	10641	VC	1001	1002	1005	1006	1011	1013	1014	1017
			1021	1022	1025	1027	1030	1033	1035	1036
			1041	1043	1044	1047	1051	1052	1055	1057
			1060	1063	1065	1066	1071	1073	1074	1077
			1101	1102	1105	1107	1110	1113	1115	1116
			1121	1123	1124	1127	1131	1132	1135	1137
			1140	1143	1145	1146	1151	1153	1154	1157
			1161	1162	1165	1167	1170	1173	1175	1176
			1201	1203	1204	1207	1211	1212	1215	1217
			1220	1223	1225	1226	1231	1233	1234	1237
			1241	1242	1245	1247	1250	1253	1255	1256
			1261	1263	1264	1267	1271	1272	1275	1277
			1300	1303	1305	1306	1311	1313	1314	1317
			1321	1322	1325	1327	1330	1333	1335	1336
			1341	1343	1344	1347	1351	1352	1355	1357
			1360	1363	1365	1366	1371	1373	1374	1377

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
		1401	1402	1405	1407	1410	1413	1415	1416
		1421	1423	1424	1427	1431	1432	1435	1437
		1440	1443	1445	1446	1451	1453	1454	1457
		1461	1462	1465	1467	1470	1473	1475	1476
		1501	1503	1504	1507	1511	1512	1515	1517
		1520	1523	1525	1526	1531	1533	1534	1537
		1541	1542	1545	1547	1550	1553	1555	1556
		1561	1563	1564	1567	1571	1572	1575	1577
		1600	1603	1605	1606	1611	1613	1614	1617
		1621	1622	1625	1627	1630	1633	1635	1636
		1641	1643	1644	1647	1651	1652	1655	1657
		1660	1663	1665	1666	1671	1673	1674	1677
		1701	1702	1705	1707	1710	1713	1715	1716
		1721	1723	1724	1727	1731	1732		
KODER	11000								
FATAL78	12200	SYSTEM	710						
CARD.F	12275	SIOS	56						
BKSPRU.	12404								
FIZBA.	12414								
POSFIL.	12642	OUTPTC	32						
RDPRU.	12732								
DAT.	12753	OUTPTC	56	72	37				
CIOI.	12560								
OPEN.	12317	INPUTC	22						
		SYSTEM	506						
		OUTPTC	22						
SIO.	12436	INPUTC	52						
		OUTPTC	71						
ADVIN.	12653	SYSTEM	410						
MVWDS.	12465								
POSFI.	12663								
FIZBA.	12675								
DAT..	13203	INPUTC	56	24	47				
GETBA	13433	INPUTC	10						
		OUTPTC	10						

UNSATISFIED EXTERNALS REFERENCES (RELATIVE)

** NONE **

VARIABLE NAME	N
PERMEABILITY	646
TRUE SPECIFIC GRAVITY	587
APPARENT SPECIFIC GRAVITY	1332
UNSPECIFIED TYPE OF SPECIFIC GRAVITY	72
POROSITY	879
ABSORPTION	301
SCLEROSCOPE HARDNESS.	321
ABRASIVE HARDNESS	118
IMPACT TOUGHNESS	192
PROPAGATION VEL-LONG WAVES/LAB	433
PROPAGATION VEL-TRANSVERSE WAVES/LAB	182
SPECIFIC DAMPING CAPACITY/LAB	203
POISSON S RATIO. DYNAMIC/LAB	392
YOUNG S MODULUS. DYNAMIC/LAB	521
MODULUS OF RIGIDITY. SHEAR MODULUS. DYNAMIC/LAB	434
SHEAR STRENGTH/STATIC	39
SHEAR STRENGTH AT SS1/STATIC	4
SHEAR STRENGTH AT SS2/STATIC	4
PROPAGATION VEL-LONG WAVES/FIELD	82
PROPAGATION VEL-TRANSVERSE WAVES/FIELD	31
SPECIFIC DAMPING CAPACITY/FIELD	27
POISSON S RATIO. DYNAMIC/FIELD	29
YOUNG S MODULUS. DYNAMIC/FIELD	61
MODULUS OF RIGIDITY. SHEAR MODULUS. DYNAMIC/FIELD	21
SHEAR STRENGTH/STATIC/FIELD	11
SHEAR STRENGTH AT SS1/STATIC/FIELD	0
SHEAR STRENGTH AT SS2/STATIC/FIELD	0
COMPRESSIVE STRENGTH. UNCONFINED/LAB	935
COMPRESSIVE STRENGTH. TRIAXIAL AT LS1/LAB	195
COMPRESSIVE STRENGTH. TRIAXIAL AT LS2/LAB	181
COMPRESSIVE STRENGTH. TRIAXIAL AT LS3/LAB	94
MODULUS OF DEFORMATION AT US1/LAB	610
POISSON S RATIO AT US1/LAB	431
YOUNG S MODULUS AT US1/LAB	392
SET FOR UPPER STRESS/LAB	0
COMPRESSIVE STRENGTH. UNCONFINED/FIELD	0
COMPRESSIVE STRENGTH. TRIAXIAL AT LS1/FIELD	37
COMPRESSIVE STRENGTH. TRIAXIAL AT LS2/FIELD	28
COMPRESSIVE STRENGTH. TRIAXIAL AT LS3/FIELD	0
MODULUS OF DEFORMATION AT US1/FIELD	57
POISSON S RATIO AT US1/FIELD	20
YOUNG S MODULUS AT US1/FIELD	20
SET FOR UPPER STRESS/FIELD	2
MODULUS OF DEFORMATION AT US2	273
POISSON S RATIO AT US2	279
YOUNG S MODULUS AT US2	159
SET FOR US2	4
MODULUS OF DEFORMATION AT US3	279
POISSON S RATIO AT US3	273
YOUNG S MODULUS AT US3	159
SET FOR US3	15
MODULUS OF DEFORMATION AT US4	189
POISSON S RATIO AT US4	195
YOUNG S MODULUS AT US4	156
SET FOR US4	133
MODULUS OF DEFORMATION AT US2/FIELD	81
POISSON S RATIO AT US2/FIELD	113
YOUNG S MODULUS AT US2/FIELD	112
SET FOR US2/FIELD	46
MODULUS OF DEFORMATION AT US30FIELD	1

POISSON'S RATIO AT US3/FIELD	0
YOUNG'S MODULUS AT US3/FIELD	0
SET FOR US3/FIELD	1
MODULUS OF DEFORMATION AT US4/FIELD	1
POISSON'S RATIO AT US4/FIELD	1
YOUNG'S MODULUS AT US4/FIELD	1
SET FOR US4/FIELD	0
TENSILE STRENGTH	918
MODULUS OF (TENSILE) DEFORMABILITY AT US1	195
POISSON'S RATIO AT US1 (TENSILE)	314
YOUNG'S MODULUS AT US1 (TENSILE)	297
MODULUS OF (TENSILE) DEFORMABILITY AT US2	466
POISSON'S RATIO AT US2 (TENSILE)	38
YOUNG'S MODULUS AT US2 (TENSILE)	324
MODULUS OF (TENSILE) DEFORMABILITY AT US3	62
POISSON'S RATIO AT US3 (TENSILE)	22
YOUNG'S MODULUS AT US3 (TENSILE)	118
MODULUS OF RUPTURE	291

MM51555. 12/19/69.PURDUE MACE 11/27/69.

```
09.16.42.MM515/     3512.NAHAS.T100,CM60000,L20000
09.16.42.*TP1=P10.
09.16.42.MAP(ON)
09.16.42.RUN(S)
09.16.48.CTIME 004.139 SEC.  RUN MOD LEVEL 4R
09.16.49.REQUEST(TAPE1,556,HI,X,C=84,MT,READ)
13.26.59.  MT53 ASSIGNED = 556
13.27.00.REWIND(TAPE1)
13.27.00.LGO(LC=20000)
13.27.02.CX     4.566 SEC.
13.27.02.PX     4.249 SEC.
13.27.02.NL    13600
13.38.05.STOP
13.38.05.CP    66.579 SEC.
13.38.05.PP   352.198 SEC.
13.38.05.LINES =          0702 OCTAL
13.38.05.CM     3.093 MWD-SEC.
```

```
                PROGRAM COUNT(INPUT,OUTPUT,TAPE1,TAPE6=OUTPUT)
000003          DIMENSION N(27),R(8),RR(8)
000003          DATA R/1H /
000003          DO 1 K=1,27
000005      1   N(K) = 0
000010          DO 1000 I=1,2170
000012          READ(1,2) A20,A19,A13,A17,A1,A6
000031      2   FORMAT(36X,A4,2X,A4,8X,A3,8X,A3,2X,A2,2X,A3)
000031          READ(1,3) B15,B16,B8,B25
000045      3   FORMAT(28X,A3,A3,2X,A3,A4)
000045          READ(1,4) C14
000053      4   FORMAT(28X,A3)
000053          READ(1,5) D22,D23,D24,D2,D9,D26,D18
000075      5   FORMAT(21X,A4,12X,A4,3X,A4,6X,A4,A3,A4,2X,A2)
000075          READ(1,6) (R(I),I=1,8)
000107      6   FORMAT(8A10)
000107          READ(1,7) F3,F10,F27,F4,F11,F5,F12
000131      7   FORMAT(10X,A4,A3,A4,11X,A4,A3,15X,A4,A3)
000131          READ(1,6) (RR(I),I=1,8)
000143          READ(1,8) H21,H7
000153      8   FORMAT(21X,A4,48X,A3)
000153     10   IF( A1.NE.R) N( 1) = N( 1) + 1
000157     11   IF( D2.NE.R) N( 2) = N( 2) + 1
000163     12   IF( F3.NE.R) N( 3) = N( 3) + 1
000167     13   IF( F4.NE.R) N( 4) = N( 4) + 1
000173     14   IF( F5.NE.R) N( 5) = N( 5) + 1
000177     15   IF( A6.NE.R) N( 6) = N( 6) + 1
000203     16   IF( H7.NE.R) N( 7) = N( 7) + 1
000207     17   IF( B8.NE.R) N( 8) = N( 8) + 1
000213     18   IF( D9.NE.R) N( 9) = N( 9) + 1
000217     19   IF(F10.NE.R) N(10) = N(10) + 1
000223     20   IF(F11.NE.R) N(11) = N(11) + 1
000227     21   IF(F12.NE.R) N(12) = N(12) + 1
000233     22   IF(A13.NE.R) N(13) = N(13) + 1
000237     23   IF(C14.NE.R) N(14) = N(14) + 1
000243     24   IF(B15.NE.R) N(15) = N(15) + 1
000247     25   IF(B16.NE.R) N(16) = N(16) + 1
000253     26   IF(A17.NE.R) N(17) = N(17) + 1
000257     27   IF(D18.NE.R) N(18) = N(18) + 1
000263     28   IF(A19.NE.R) N(19) = N(19) + 1
000267     29   IF(A20.NE.R) N(20) = N(20) + 1
000273     30   IF(H21.NE.R) N(21) = N(21) + 1
000277     31   IF(D22.NE.R) N(22) = N(22) + 1
000303     32   IF(D23.NE.R) N(23) = N(23) + 1
000307     33   IF(D24.NE.R) N(24) = N(24) + 1
000313     34   IF(B25.NE.R) N(25) = N(25) + 1
000317     35   IF(D26.NE.R) N(26) = N(26) + 1
000323     36   IF(F27.NE.R) N(27) = N(27) + 1
000327   1000   CONTINUE
000331          WRITE(6,909)
000335    909   FORMAT(1H1)
000335          WRITE(6,999) ((K,N(K)),K=1,27)
000350    999   FORMAT(1H ,2HN(,I2,4H) = ,I4)
000350          STOP
000352          END


PROGRAM LENGTH INCLUDING I/O BUFFERS
003436

UNUSED COMPILER SPACE
020700
```

Program No. 1A

COUNT PROGRAM

Number of non-blank entries for selected variables on a tape

FWA LOAD 100 LWA LOAD 10743 FWA LOADER 57303 FWA TABLES 57021 UNUSED STORAGE 46056

PROGRAM	ADDRESS	FILE
COUNT	100	LGO
INPUTC	3736	SYSTEM
SYSTEM	5057	SYSTEM
OUTPTC	6130	SYSTEM
FATAL7R	7470	SYSTEM
SIO$	7606	SYSTEM
GETRA	10724	SYSTEM

COMMON	ADDRESS	LENGTH
SCOPE2	5057	
/BLANK/	0	0

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
COUNT	101								
INPUTC	3740	COUNT	13	15	17	21	23	25	27	30
			33	35	37	41	43	44	47	51
			52	55	57	61	63	65	67	71
			73	74	77	103	106	111	113	115
			117	121	123	125	127	130	133	137
			142	145	147	151	152			
KRAKER	4042									
OENTRY	5060	COUNT	2							
SYSTEM	5265	INPUTC	37	727						
		OUTPTC	74	1144						
SYSTEMC	5231									
SYSTEMB	5260									
END	5154	COUNT	353							
STOP	5204	COUNT	351							
EXIT	5176									
ABNORMI	5214	INPUTC	40	730						
		OUTPTC	15	1145						
OUTPTC	6132	COUNT	333	334	337	342	344	347		
KODER	6271									
FATAL7R	7471	SYSTEM	710							
CARD.F	7566	SIO$	56							
BKSPRU.	10075									
FIZRAK.	10105									
POSFIL.	10133	OUTPTC	32							
RDPRU.	10223									
DAT.	10244	OUTPTC	56	72	37					
CIO1.	10051									
OPEN.	7610	INPUTC	22							
		SYSTEM	506							
		OUTPTC	22							
SIO.	7727	INPUTC	52							
		OUTPTC	71							
ADVIN.	10144	SYSTEM	410							
MVWDS.	7756									
POSFI.	10154									
FIZRA.	10166									
DAT..	10474	INPUTC	56	24	47					
GETRA	10724	INPUTC	10							
		OUTPTC	10							

UNSATISFIED EXTERNALS REFERENCES (RELATIVE)

** NONE **

```
N( 1) =  115
N( 2) =  584
N( 3) =  266
N( 4) =  272
N( 5) =  186
N( 6) =  184
N( 7) =  266
N( 8) =  385
N( 9) =  417
N(10) =  272
N(11) =  266
N(12) =  193
N(13) =  867
N(14) =   88
N(15) =  423
N(16) =  178
N(17) =  310
N(18) =    0
N(19) = 1322
N(20) =  578
N(21) =  994
N(22) =  877
N(23) =  175
N(24) =   60
N(25) =  490
N(26) =  389
N(27) =  159
```

```
MM5165A.   12/19/69.PURDUE MACE 11/27/69.

09.28.43.MM516/     3512,NAHAS,T180,CM60000,TP1,P1
09.28.43.0.
09.28.43.MAP(ON)
09.28.43.RUN(S)
09.28.46.CTIME 000.625 SEC.  RUN MOD LEVEL 4B
09.28.47.REQUEST(TAPE1,556,HY,X,C=84,MT,READ)
14.06.01.  MT51 ASSIGNED - 556
14.06.01.REWIND(TAPE1)
14.06.01.LGO.
14.06.04.CX      .847 SEC.
14.06.04.PX     3.584 SEC.
14.06.04.NL    11100
14.26.21.STOP
14.26.21.CP    37.450 SEC.
14.26.21.PP   324.028 SEC.
14.26.21.LINES =          0222 OCTAL
14.26.21.CM     3.564 MWD-SEC.
```

Program No. 2

COUNT PROGRAM

Number of non-blank entries for both variables in the same data set for selected pairs of variables

```
      PROGRAM COUNT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
000003      DIMENSION N(136)
000003      DATA B/1H /
000033      DO 1 K=1,136
000035    1 N(K) = 0
000040      DO 100 K=1,2170
000012      READ(1,2) A1,A2,A3
000023    2 FORMAT(32X,A1,15X,A4,7X,A4)
000023      READ(1,3) B4,B5,B6
000035    3 FORMAT(34X,A2,7X,A4,14X,A3)
000035      READ(1,4) C7
000043    4 FORMAT(39X,A4)
000043      READ(1,5) D8,D9
000053    5 FORMAT(30X,A4,20X,A4)
000053      READ(1,6) E10,E11
000063    6 FORMAT(54X,A4,3X,A4)
000063      READ(1,7) F12,F13,F14
000075    7 FORMAT(39X,A4,18X,A4,2X,A2)
000075      READ(1,8) H
000103    8 FORMAT(20X,A4)
000103      READ(1,9) H15,H16,H17
000115    9 FORMAT(31X,A4,10X,A4,10X,A4)
000115      IF(( A1.NE.B ).AND.( A2.NE.B )) N( 1) = N( 1) + 1
000125      IF(( A1.NE.B ).AND.( A3.NE.B )) N( 2) = N( 2) + 1
000137      IF(( A1.NE.B ).AND.( B4.NE.B )) N( 3) = N( 3) + 1
000150      IF(( A1.NE.B ).AND.( B5.NE.B )) N( 4) = N( 4) + 1
000161      IF(( A1.NE.B ).AND.( B6.NE.B )) N( 5) = N( 5) + 1
000172      IF(( A1.NE.B ).AND.( C7.NE.B )) N( 6) = N( 6) + 1
000203      IF(( A1.NE.B ).AND.(D8 .NE.B )) N( 7) = N( 7) + 1
000214      IF(( A1.NE.B ).AND.(D9 .NE.B )) N( 8) = N( 8) + 1
000225      IF(( A1.NE.B ).AND.(E10.NE.B )) N( 9) = N( 9) + 1
000236      IF(( A1.NE.B ).AND.(E11.NE.B )) N(10) = N(10) + 1
000247      IF(( A1.NE.B ).AND.(F12.NE.B )) N(11) = N(11) + 1
000260      IF(( A1.NE.B ).AND.(F13.NE.B )) N(12) = N(12) + 1
000271      IF(( A1.NE.B ).AND.(F14.NE.B )) N(13) = N(13) + 1
000302      IF(( A1.NE.B ).AND.(H15.NE.B )) N(14) = N(14) + 1
000313      IF(( A1.NE.B ).AND.(H16.NE.B )) N(15) = N(15) + 1
000324      IF(( A1.NE.B ).AND.(H17.NE.B )) N(16) = N(16) + 1
000335      IF(( A3.NE.B ).AND.( B4.NE.B )) N(18) = N(18) + 1
000346      IF(( A3.NE.B ).AND.( B5.NE.B )) N(19) = N(19) + 1
000357      IF(( A3.NE.B ).AND.( B6.NE.B )) N(20) = N(20) + 1
000370      IF(( A3.NE.B ).AND.( C7.NE.B )) N(21) = N(21) + 1
000401      IF(( A3.NE.B ).AND.( D8.NE.B )) N(22) = N(22) + 1
000412      IF(( A3.NE.B ).AND.( D9.NE.B )) N(23) = N(23) + 1
000423      IF(( A3.NE.B ).AND.(E10.NE.B )) N(24) = N(24) + 1
000434      IF(( A3.NE.B ).AND.(E11.NE.B )) N(25) = N(25) + 1
000445      IF(( A3.NE.B ).AND.(F12.NE.B )) N(26) = N(26) + 1
000456      IF(( A3.NE.B ).AND.(F13.NE.B )) N(27) = N(27) + 1
000467      IF(( A3.NE.B ).AND.(F14.NE.B )) N(28) = N(28) + 1
000500      IF(( A3.NE.B ).AND.(H15.NE.B )) N(29) = N(29) + 1
000511      IF(( A3.NE.B ).AND.(H16.NE.B )) N(30) = N(30) + 1
000522      IF(( A3.NE.B ).AND.(H17.NE.B )) N(31) = N(31) + 1
000533      IF(( A3.NE.B ).AND.( A2.NE.B )) N(32) = N(32) + 1
000544      IF(( A2.NE.B ).AND.( B4.NE.B )) N(33) = N(33) + 1
000555      IF(( A2.NE.B ).AND.( B5.NE.B )) N(34) = N(34) + 1
000566      IF(( A2.NE.B ).AND.( B6.NE.B )) N(35) = N(35) + 1
000577      IF(( A2.NE.B ).AND.( C7.NE.B )) N(36) = N(36) + 1
000610      IF(( A2.NE.B ).AND.( D8.NE.B )) N(37) = N(37) + 1
000621      IF(( A2.NE.B ).AND.( D9.NE.B )) N(38) = N(38) + 1
000632      IF(( A2.NE.B ).AND.(E10.NE.B )) N(39) = N(39) + 1
000643      IF(( A2.NE.B ).AND.(E11.NE.B )) N(40) = N(40) + 1
000654      IF(( A2.NE.B ).AND.(F12.NE.B )) N(41) = N(41) + 1
000665      IF(( A2.NE.B ).AND.(F13.NE.B )) N(42) = N(42) + 1
000676      IF(( A2.NE.B ).AND.(F14.NE.B )) N(43) = N(43) + 1
000707      IF(( A2.NE.B ).AND.(H15.NE.B )) N(44) = N(44) + 1
000720      IF(( A2.NE.B ).AND.(H16.NE.B )) N(45) = N(45) + 1
000731      IF(( A2.NE.B ).AND.(H17.NE.B )) N(17) = N(17) + 1
000742      IF(( B4.NE.B ).AND.( B5.NE.B )) N(46) = N(46) + 1
000753      IF(( B4.NE.B ).AND.( B6.NE.B )) N(47) = N(47) + 1
000764      IF(( B4.NE.B ).AND.(C7 .NE.B )) N(48) = N(48) + 1
000775      IF(( B4.NE.B ).AND.( D8.NE.B )) N(49) = N(49) + 1
001006      IF(( B4.NE.B ).AND.( D9.NE.B )) N(50) = N(50) + 1
001017      IF(( B4.NE.B ).AND.(E10.NE.B )) N(51) = N(51) + 1
001030      IF(( B4.NE.B ).AND.(E11.NE.B )) N(52) = N(52) + 1
001041      IF(( B4.NE.B ).AND.(F12.NE.B )) N(53) = N(53) + 1
001052      IF(( B4.NE.B ).AND.(F13.NE.B )) N(54) = N(54) + 1
001063      IF(( B4.NE.B ).AND.(F14.NE.B )) N(55) = N(55) + 1
001074      IF(( B4.NE.B ).AND.(H15.NE.B )) N(56) = N(56) + 1
001105      IF(( B4.NE.B ).AND.(H16.NE.B )) N(57) = N(57) + 1
001116      IF(( B4.NE.B ).AND.(H17.NE.B )) N(58) = N(58) + 1
001127      IF(( B5.NE.B ).AND.( B6.NE.B )) N(59) = N(59) + 1
001140      IF(( B5.NE.B ).AND.( C7.NE.B )) N(60) = N(60) + 1
001151      IF(( B5.NE.B ).AND.( D8.NE.B )) N(61) = N(61) + 1
001162      IF(( B5.NE.B ).AND.( D9.NE.B )) N(62) = N(62) + 1
001173      IF(( B5.NE.B ).AND.(E10.NE.B )) N(63) = N(63) + 1
001204      IF(( B5.NE.B ).AND.(E11.NE.B )) N(64) = N(64) + 1
001215      IF(( B5.NE.B ).AND.(F12.NE.B )) N(65) = N(65) + 1
001226      IF(( B5.NE.B ).AND.(F13.NE.B )) N(66) = N(66) + 1
001237      IF(( B5.NE.B ).AND.(F14.NE.B )) N(67) = N(67) + 1
001250      IF(( B5.NE.B ).AND.(H15.NE.B )) N(68) = N(68) + 1
001261      IF(( B5.NE.B ).AND.(H16.NE.B )) N(69) = N(69) + 1
001272      IF(( B5.NE.B ).AND.(H17.NE.B )) N(70) = N(70) + 1
001303      IF(( B6.NE.B ).AND.( C7.NE.B )) N(71) = N(71) + 1
001314      IF(( B6.NE.B ).AND.( D8.NE.B )) N(72) = N(72) + 1
001325      IF(( B6.NE.B ).AND.( D9.NE.B )) N(73) = N(73) + 1
001336      IF(( B6.NE.B ).AND.(E10.NE.B )) N(74) = N(74) + 1
001347      IF(( B6.NE.B ).AND.(E11.NE.B )) N(75) = N(75) + 1
001360      IF(( B6.NE.B ).AND.(F12.NE.B )) N(76) = N(76) + 1
001371      IF(( B6.NE.B ).AND.(F13.NE.B )) N(77) = N(77) + 1
001402      IF(( B6.NE.B ).AND.(F14.NE.B )) N(78) = N(78) + 1
001413      IF(( B6.NE.B ).AND.(H15.NE.B )) N(79) = N(79) + 1
001424      IF(( B6.NE.B ).AND.(H16.NE.B )) N(80) = N(80) + 1
001435      IF(( B6.NE.B ).AND.(H17.NE.B )) N(81) = N(81) + 1
001446      IF(( C7.NE.B ).AND.( D8.NE.B )) N(82) = N(82) + 1
001457      IF(( C7.NE.B ).AND.( D9.NE.B )) N(83) = N(83) + 1
001470      IF(( C7.NE.B ).AND.(E10.NE.B )) N(84) = N(84) + 1
001501      IF(( C7.NE.B ).AND.(E11.NE.B )) N(85) = N(85) + 1
001512      IF(( C7.NE.B ).AND.(F12.NE.B )) N(86) = N(86) + 1
001523      IF(( C7.NE.B ).AND.(F13.NE.B )) N(87) = N(87) + 1
001534      IF(( C7.NE.B ).AND.(F14.NE.B )) N(88) = N(88) + 1
001545      IF(( C7.NE.B ).AND.(H15.NE.B )) N(89) = N(89) + 1
001556      IF(( C7.NE.B ).AND.(H16.NE.B )) N(90) = N(90) + 1
001567      IF(( C7.NE.B ).AND.(H17.NE.B )) N(91) = N(91) + 1
001600      IF(( D8.NE.B ).AND.( D9.NE.B )) N(92) = N(92) + 1
001611      IF(( D8.NE.B ).AND.(E10.NE.B )) N(93) = N(93) + 1
001622      IF(( D8.NE.B ).AND.(E11.NE.B )) N(94) = N(94) + 1
001633      IF(( D8.NE.B ).AND.(F12.NE.B )) N(95) = N(95) + 1
001644      IF(( D8.NE.B ).AND.(F13.NE.B )) N(96) = N(96) + 1
001655      IF(( D8.NE.B ).AND.(F14.NE.B )) N(97) = N(97) + 1
001666      IF(( D8.NE.B ).AND.(H15.NE.B )) N(98) = N(98) + 1
001677      IF(( D8.NE.B ).AND.(H16.NE.B )) N(99) = N(99) + 1
```

```
001710          IF(( D8.NE.B ).AND.(H17.NE.B )) N(100) = N(100) + 1
001721          IF(( D9.NE.B ).AND.(E10.NE.B )) N(101) = N(101) + 1
001732          IF(( D9.NE.B ).AND.(E11.NE.B )) N(102) = N(102) + 1
001743          IF(( D9.NE.B ).AND.(F12.NE.B )) N(103) = N(103) + 1
001754          IF(( D9.NE.B ).AND.(F13.NE.B )) N(104) = N(104) + 1
001765          IF(( D9.NE.B ).AND.(F14.NE.B )) N(105) = N(105) + 1
001776          IF(( D9.NE.B ).AND.(H15.NE.B )) N(106) = N(106) + 1
002007          IF(( D9.NE.B ).AND.(H16.NE.B )) N(107) = N(107) + 1
002020          IF(( D9.NE.B ).AND.(H17.NE.B )) N(108) = N(108) + 1
002031          IF((E10.NE.B ).AND.(E11.NE.B )) N(109) = N(109) + 1
002042          IF((E10.NE.B ).AND.(F12.NE.B )) N(110) = N(110) + 1
002053          IF((E10.NE.B ).AND.(F13.NE.B )) N(111) = N(111) + 1
002064          IF((E10.NE.B ).AND.(F14.NE.B )) N(112) = N(112) + 1
002075          IF((E10.NE.B ).AND.(H15.NE.B )) N(113) = N(113) + 1
002106          IF((E10.NE.B ).AND.(H16.NE.B )) N(114) = N(114) + 1
002117          IF((E10.NE.B ).AND.(H17.NE.B )) N(115) = N(115) + 1
002130          IF((E11.NE.B ).AND.(F12.NE.B )) N(116) = N(116) + 1
002141          IF((E11.NE.B ).AND.(F13.NE.B )) N(117) = N(117) + 1
002152          IF((E11.NE.B ).AND.(F14.NE.B )) N(118) = N(118) + 1
002163          IF((E11.NE.B ).AND.(H15.NE.B )) N(119) = N(119) + 1
002174          IF((E11.NE.B ).AND.(H16.NE.B )) N(120) = N(120) + 1
002205          IF((E11.NE.B ).AND.(H17.NE.B )) N(121) = N(121) + 1
002216          IF((F12.NE.B ).AND.(F13.NE.B )) N(122) = N(122) + 1
002227          IF((F12.NE.B ).AND.(F14.NE.B )) N(123) = N(123) + 1
002240          IF((F12.NE.B ).AND.(H15.NE.B )) N(124) = N(124) + 1
002251          IF((F12.NE.B ).AND.(H16.NE.B )) N(125) = N(125) + 1
002262          IF((F12.NE.B ).AND.(H17.NE.B )) N(126) = N(126) + 1
002273          IF((F13.NE.B ).AND.(F14.NE.B )) N(127) = N(127) + 1
002304          IF((F13.NE.B ).AND.(H15.NE.B )) N(128) = N(128) + 1
002315          IF((F13.NE.B ).AND.(H16.NE.B )) N(129) = N(129) + 1
002326          IF((F13.NE.B ).AND.(H17.NE.B )) N(130) = N(130) + 1
002337          IF((F14.NE.B ).AND.(H15.NE.B )) N(131) = N(131) + 1
002347          IF((F14.NE.B ).AND.(H16.NE.B )) N(132) = N(132) + 1
002357          IF((F14.NE.B ).AND.(H17.NE.B )) N(133) = N(133) + 1
002367          IF((H15.NE.B ).AND.(H16.NE.B )) N(134) = N(134) + 1
002377          IF((H15.NE.B ).AND.(H17.NE.B )) N(135) = N(135) + 1
002407          IF((H16.NE.B ).AND.(H17.NE.B )) N(136) = N(136) + 1
002417      100 CONTINUE
002421          WRITE(6,900)
002425      900 FORMAT(1H1)
002425          WRITE(6,910) ((K,N(K)),K=1,136)
002440      910 FORMAT(1H ,2HN(,I2,4H) = ,I4)
002440          STOP
002442          END

PROGRAM LENGTH INCLUDING I/O BUFFERS
007110

UNUSED COMPILER SPACE


011700
```

LOAD MAP FILE = LGO 13.00.04. 12/19/69. PAGE 1

FWA LOAD 100 LWA LOAD 14215 FWA LOADER 57303 FWA TABLES 57026 UNUSED STORAGE 4241

PROGRAM	ADDRESS	FILE
COUNT	100	LGO
INPUTC	7210	SYSTEM
SYSTEM	10331	SYSTEM
OUTPTC	11402	SYSTEM
FATAL7B	12742	SYSTEM
SIOS	13060	SYSTEM
BETBA	14176	SYSTEM

COMMON	ADDRESS	LENGTH
SCOPE2	10331	
/BLANK/	0	0

ENTRY	ADDRESS
COUNT	101
INPUTC	7212
KRAKER	7314
GENTRY	10332
SYSTEM	10537
SYSTEMC	10503
SYSTEMD	10532
END	10426
STOP	10456
EXIT	10450
ABNORM	10466
OUTPTC	11404
KODER	11543
FATAL7B	12743
C400.F	13040
RKSPRU.	13347
FIZBAK.	13357
POSFIL.	13405
RDPRU.	13475
DAT.	13516
CIOI.	13323
OPEN.	13062
SIO.	13201
ADVIN.	13416
MVWDS.	13230
POSFI.	13426
FIZBA.	13440
DAT..	13746
BETBA	14176

REFERENCES (RELATIVE)

COUNT	13	15	17	21	22	25	27	31
	33	34	37	41	42	45	47	51
	52	55	57	61	62	65	67	71
	73	74	77	101	102	105	107	111
	113	114						
COUNT	2							
INPUTC	37	727						
OUTPTC	14	1144						
COUNT	2443							
COUNT	2441							
INPUTC	40	730						
OUTPTC	15	1145						
COUNT	2423	2424	2427	2432	2436	2437		
SYSTEM	710							
SIOS	56							
OUTPTC	32							
OUTPTC	56	72	37					
INPUTC	22							
SYSTEM	506							
OUTPTC	22							
INPUTC	52							
OUTPTC	71							
SYSTEM	410							
INPUTC	56	24	47					
INPUTC	10							
OUTPTC	10							

LOAD MAP FILE = LGO 13.00.04. 12/19/69. PAGE 2

UNSATISFIED EXTERNALS REFERENCES (RELATIVE)

** NONE **

```
N( 1) =   28
N( 2) =  137
N( 3) =  189
N( 4) =  343
N( 5) =    0
N( 6) =    2
N( 7) =  148
N( 8) =  268
N( 9) =    0
N(10) =    0
N(11) =  109
N(12) =  108
N(13) =   85
N(14) =   21
N(15) =   21
N(16) =   21
N(17) =    0
N(18) =    0
N(19) =   33
N(20) =    0
N(21) =    0
N(22) =   90
N(23) =  191
N(24) =    0
N(25) =    0
N(26) =  100
N(27) =   99
N(28) =   90
N(29) =    0
N(30) =    0
N(31) =    0
N(32) =   23
N(33) =    2
N(34) =   16
N(35) =    0
N(36) =    0
N(37) =   20
N(38) =   48
N(39) =    0
N(40) =    0
N(41) =   19
N(42) =   18
N(43) =   18
N(44) =    0
N(45) =    0
N(46) =  255
N(47) =    0
N(48) =    1
N(49) =    0
N(50) =   74
N(51) =    0
N(52) =    0
N(53) =    1
N(54) =    1
N(55) =    0
N(56) =   21
N(57) =   21
N(58) =   21
N(59) =    0
N(60) =    7
N(61) =   51
N(62) =  149
N(63) =    0
N(64) =    0
N(65) =   39
N(66) =   37
N(67) =   31
N(68) =   21
N(69) =   21
N(70) =   21
N(71) =    0
N(72) =    0
N(73) =   12
N(74) =    0
N(75) =    0
N(76) =    0
N(77) =    0
N(78) =    0
N(79) =    0
N(80) =    0
N(81) =    0
N(82) =    0
N(83) =   18
N(84) =    3
N(85) =    1
N(86) =    0
N(87) =    0
N(88) =    0
N(89) =    0
N(90) =    0
N(91) =    0
N(92) =  193
N(93) =    0
N(94) =    0
N(95) =   93
N(96) =   91
N(97) =   79
N(98) =    0
N(99) =    0
N(100) =   0
N(101) =   2
N(102) =   2
N(103) =  150
N(104) =  147
N(105) =  135
N(106) =   22
N(107) =   22
N(108) =   22
N(109) =   41
N(110) =    0
N(111) =    0
N(112) =    0
N(113) =    0
N(114) =    0
N(115) =    0
N(116) =    0
N(117) =    0
N(118) =    0
N(119) =    0
N(120) =    0
N(121) =    0
N(122) =  156
N(123) =  134
N(124) =    0
N(125) =    0
N(126) =    0
N(127) =  132
N(128) =    0
N(129) =    0
N(130) =    0
N(131) =    0
N(132) =    0
N(133) =    0
N(134) =   25
N(135) =   25
N(136) =   25
```



```
MM51149.   12/19/69.PURDUE MACE 11/27/69.

09.29.40.MM511/     3512,NAHAS,T240,CM60000,TP1,P1
09.29.40.0.
09.29.40.MAP(ON)
09.29.40.RUN(S)
09.29.46.CTIME 003.639 SEC.  RUN MOD LEVEL 48
09.29.47.REQUEST(TAPE1,556,HY,X,C=84,MT,READ)
12.59.59.  MT53 ASSIGNED - 556
12.59.59.REWIND(TAPE1)
13.00.01.LGO.
13.00.04.CX     3.851 SEC.
13.00.04.PX     4.230 SEC.
13.00.04.NL    14400
13.10.57.STOP
13.10.57.CP    37.776 SEC.
13.10.58.PP   313.462 SEC.
13.10.58.LINES =          0554 OCTAL
13.10.58.CM     5.784 MWD-SEC.
```

Program No. 3

Program to print out all values within specified ranges (and associated rock types) for specific selection of variables

```
                PROGRAM MAIN(INPUT,OUTPUT,PUNCH,TAPE1,TAPE5=INPUT,TAPE6=OUTPUT,
               1  TAPE7=PUNCH)
000003          DATA X/1H /
000003          DIMENSION CARD(9)
000003          DIMENSION PERM(1000),SGT(700),SGA(1000),SGU(300),PORO(1300)
000003          DIMENSION ABSOR(500),SCLERO(500),ABRAS(200),TOUGH(300)
000003          DIMENSION RT1(1000),RT2(700),RT3(1600),RT4(300),RT5(1300)
000003          DIMENSION RT6(500),RT7(500),RT8(200),RT9(300)
000003          INTEGER RT1,RT2,RT3,RT4,RT5,RT6,RT7,RT8,RT9
000003          INTEGER A2,A3
000003          REAL MEAN,MAX,MIN

C       RT = ROCK TYPE  (16-18)
C  1. PERM = PERMEABILITY  (33)
C  2. SGT = TRUE SPECIFIC GRAVITY (A20)  (37-40)
C  3. SGA = APPARENT SPECIFIC GRAVITY (A19)  (43-46)
C  4. SGU = UNSPECIFIED TYPE OF SPECIFIC GRAVITY  (49-52)
C  5. PORO = POROSITY (A13)  (55-57)
C  6. ABSOR = ABSORPTION  (60-63)
C  7. SCLERO = SCLEROSCOPIC HARDNESS (A17)  (66-68)
C  8. ABRAS = ABRASIVE HARDNESS (A1)  (71-72)
C  9. TOUGH = IMPACT TOUGHNESS (A6)  (75-77)
000003          N1=0
000004          N2=0
000004          N3=0
000005          N4=0
000006          N5=0
000007          N6=0
000010          N7=0
000011          N8=0
000012          N9=0
000014          DO 100 K=1,2170
000015          READ(1,500) CARD
000022      500 FORMAT(9A10)
000022          DECODE(80,2,CARD(1)) IRT,A1,A2,A3,A4,A5,A6,A7,A8,A9
000054      2   FORMAT(I5X,I3,1X,A1,3X,I4,2X,I4,2X,A4,2X,A3,2X,A4,2X,A3,2X,A2,2X,
               1A3)
000054          READ(1,3) B
000062          READ(1,3) C
000070          READ(1,3) D
000076          READ(1,3) E
000104          READ(1,3) F
000112          READ(1,3) G
000120          READ(1,3) H
000126        3 FORMAT(21XA4)
000126       11 IF(A1.NE.X) GO TO 31
000130       12 IF((A2.LT.6400).AND.(A2.GT.4777)) GO TO 32
000140       13 IF((A3.LT.2026).AND.(A3.GT.48)) GO TO 33
000150       14 IF(A4.NE.X) GO TO 34
000152       15 IF(A5.NE.X) GO TO 35
000154       16 IF(A6.NE.X) GO TO 36
000156       17 IF(A7.NE.X) GO TO 37
000160       18 IF(A8.NE.X) GO TO 38
000162       19 IF(A9.NE.X) GO TO 39
000164          GO TO 100
000165       31 N1=N1+1
000167          DECODE(80,40,CARD(1)) PERM(N1)
000176       40 FORMAT(32X,F1.0)
000176          RT1(N1)=IRT
000200          GO TO 12
000201       32 N2=N2+1
000203          DECODE(80,41,CARD(1)) SGT(N2)
000212       41 FORMAT(36X,F4.0)
000212          RT2(N2)=IRT
000214          GO TO 13
000215       33 N3=N3+1
000217          DECODE(80,42,CARD(1)) SGA(N3)
000226       42 FORMAT(42X,F4.0)
000226          RT3(N3)=IRT
000230          GO TO 14
000231       34 N4=N4+1
000233          DECODE(80,43,CARD(1)) SGU(N4)
000242       43 FORMAT(48X,F4.0)
000242          RT4(N4)=IRT
000244          GO TO 15
000245       35 N5=N5+1
000247          DECODE(80,44,CARD(1)) PORO(N5)
000256       44 FORMAT(54X,F3.0)
000256          RT5(N5)=IRT
000260          GO TO 16
000261       36 N6=N6+1
000263          DECODE(80,45,CARD(1)) ABSOR(N6)
000272       45 FORMAT(59X,F4.0)
000272          RT6(N6)=IRT
000274          GO TO 17
000275       37 N7=N7+1
000277          DECODE(80,46,CARD(1)) SCLERO(N7)
000306       46 FORMAT(65X,F3.0)
000306          RT7(N7)=IRT
000310          GO TO 18
000311       38 N8=N8+1
000313          DECODE(80,47,CARD(1)) ABRAS(N8)
000322       47 FORMAT(70X,F2.0)
000322          RT8(N8)=IRT
000324          GO TO 19
000325       39 N9=N9+1
000327          DECODE(80,48,CARD(1)) TOUGH(N9)
000336       48 FORMAT(74X,F3.0)
000336          RT9(N9)=IRT
000340      100 CONTINUE
```

```
000342          WRITE(6,51)
000346       51 FORMAT(1H1,40X,12HPERMEABILITY//)
000346          WRITE(6,60)   (PERM(I),RT1(I), I=1,N1)
000363       60 FORMAT(10(F6.1,2H (,I3,1H)))
000363          WRITE(6,70)
000367       70 FORMAT(/10X,44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000367          CALL MMM (N1,PERM,MEAN,MAX,MIN)
000373          WRITE(6,61) N1,MEAN,MAX,MIN
000407       61 FORMAT(///5X,3HN =I6/2X6HMEAN =F8.0/3X5HMAX =F8.0/3XEHMIN =F6.0)

000407          WRITE(6,52)
000413       52 FORMAT(1H1,40X,21HTRUE SPECIFIC GRAVITY//)
000413          WRITE(6,60)   (SG1(I),RT2(I), I=1,N2)
000430          WRITE(6,70)
000434          CALL MMM (N2,SG1,MEAN,MAX,MIN)
000440          WRITE(6,61) N2,MEAN,MAX,MIN

000454          WRITE(6,53)
000460       53 FORMAT(1H1,40X,25HAPPARENT SPECIFIC GRAVITY//)
000460          WRITE(6,60)   (SGA(I),RT3(I), I=1,N3)
000475          WRITE(6,70)
000501          CALL MMM (N3,SGA,MEAN,MAX,MIN)
000505          WRITE(6,61) N3,MEAN,MAX,MIN

000521          WRITE(6,54)
000525       54 FORMAT(1H1,40X,36HQUASI-EFFECTIVE TYPE OF SPECIFIC GRAVITY//)
000525          WRITE(6,60)   (SGU(I),RT4(I), I=1,N4)
000542          WRITE(6,70)
000546          CALL MMM (N4,SGU,MEAN,MAX,MIN)
000552          WRITE(6,61) N4,MEAN,MAX,MIN

000566          WRITE(6,55)
000572       55 FORMAT(1H1,40X,8HPOROSITY//)
000572          WRITE(6,60)   (PORO(I),RT5(I), I=1,N5)
000607          WRITE(6,70)
000613          CALL MMM (N5,PORO,MEAN,MAX,MIN)
000617          WRITE(6,61) N5,MEAN,MAX,MIN

000633          WRITE(6,56)
000637       56 FORMAT(1H1,40X,10HABSORPTION//)
000637          WRITE(6,60) (ABSOP(I),RT6(I), I=1,N6)
000654          WRITE(6,70)
000660          CALL MMM (N6,ABSOP,MEAN,MAX,MIN)
000664          WRITE(6,61) N6,MEAN,MAX,MIN

000700          WRITE(6,57)
000704       57 FORMAT(1H1,40X,21HSCLEROSCOPIC HARDNESS//)
000704          WRITE(6,60)(SCLERO(I),RT7(I), I=1,N7)
000721          WRITE(6,70)
000725          CALL MMM (N7,SCLERO,MEAN,MAX,MIN)
000731          WRITE(6,61) N7,MEAN,MAX,MIN

000745          WRITE(6,58)
000751       58 FORMAT(1H1,40X,17HABRASIVE HARDNESS//)
000751          WRITE(6,60) (ABRAS(I),RT8(I), I=1,N8)
000766          WRITE(6,70)
000772          CALL MMM (N8,ABRAS,MEAN,MAX,MIN)
000776          WRITE(6,61) N8,MEAN,MAX,MIN

001012          WRITE(6,59)
001016       59 FORMAT(1H1,40X,16HIMPACT TOUGHNESS//)
001016          WRITE(6,60) (TOUGH(I),RT9(I), I=1,N9)
001033          WRITE(6,70)
001037          CALL MMM (N9,TOUGH,MEAN,MAX,MIN)
001043          WRITE(6,61) N9,MEAN,MAX,MIN

001057          END
```

PROGRAM LENGTH INCLUDING I/O BUFFERS

036440

UNUSED COMPILER SPACE
024200

```
                SUBROUTINE MMM(N,X,MEAN,MAX,MIN)
000010          DIMENSION X(2000)
000010          REAL MEAN,MAX,MIN
000010          SUM=0.
000010          MAX=0.
000010          MIN=10000.0
000011          DO 100 I=1,N
000014          SUM=SUM+X(I)
000016          IF(X(I).GE.MAX) MAX=X(I)
000023      100 IF(X(I).LE.MIN) MIN=X(I)
000032          MEAN=SUM/FLOAT(N)
000034          RETURN
000034          END
```

SUBPROGRAM LENGTH
000051

UNUSED COMPILER SPACE
027500

49 (3)	0 (3)	17 (5)	57 (7)	57 (7)	100 (7)	1 (7)	2 (7)	14 (11)	1 (13)
4 (13)	6 (14)	9 (21)	4 (21)	6 (21)	1 (21)	1 (22)	1 (22)	6 (22)	3 (22)
1 (22)	4 (22)	4 (22)	1 (22)	3 (22)	2 (22)	2 (22)	3 (22)	4 (22)	16 (22)
4 (22)	5 (22)	4 (22)	3 (22)	3 (22)	2 (22)	11 (22)	2 (22)	2 (22)	3 (22)
6 (22)	4 (22)	4 (22)	3 (22)	9 (22)	4 (22)	2 (23)	2 (24)	2 (24)	3 (40)
0 (40)	2 (45)	8 (45)	2 (45)	1 (45)	3 (45)	3 (45)	3 (45)	4 (45)	7 (45)
17 (48)	6 (52)	2 (64)	165 (208)	165 (208)	4 (208)	3 (208)	7 (208)	83 (208)	17 (211)
13 (212)	478 (214)	18 (215)	5 (215)	75 (218)	18 (218)	50 (218)	46 (218)	46 (218)	1 (220)
60 (223)	71 (223)	200 (223)	200 (223)	178 (223)	200 (223)	12 (223)	12 (223)	10 (223)	21 (223)
4 (223)	8 (223)	18 (223)	19 (223)	2 (223)	64 (223)	81 (223)	68 (223)	111 (223)	75 (223)
78 (223)	66 (223)	111 (223)	129 (223)	80 (223)	89 (223)	108 (223)	77 (223)	109 (223)	34 (223)
63 (223)	84 (223)	5 (223)	3 (223)	16 (223)	1 (223)	5 (223)	8 (223)	2 (223)	3 (223)
7 (223)	60 (223)	17 (223)	1 (223)	7 (223)	1 (223)	13 (223)	3 (223)	4 (223)	39 (223)
6 (223)	96 (223)	61 (223)	1 (228)	76 (236)	89 (236)	89 (236)	42 (236)	23 (236)	30 (236)
18 (236)	125 (236)	110 (236)	94 (236)	152 (236)	127 (236)	119 (236)	147 (236)	98 (236)	111 (236)
122 (236)	107 (236)	142 (236)	106 (236)	111 (236)	115 (236)	103 (236)	118 (236)	120 (236)	108 (236)
120 (236)	111 (236)	130 (236)	125 (236)	127 (236)	127 (236)	127 (236)	108 (236)	120 (236)	110 (236)
110 (236)	110 (236)	110 (236)	110 (236)	120 (236)	120 (236)	120 (236)	120 (236)	120 (236)	132 (236)
126 (236)	126 (236)	126 (236)	126 (236)	126 (236)	97 (236)	97 (236)	127 (236)	127 (236)	104 (236)
133 (236)	101 (236)	121 (236)	142 (236)	133 (236)	126 (236)	126 (236)	133 (236)	131 (236)	135 (236)
131 (236)	127 (236)	121 (236)	102 (236)	126 (236)	137 (236)	132 (236)	123 (236)	130 (236)	130 (236)
135 (236)	114 (236)	112 (236)	116 (236)	113 (236)	117 (236)	130 (236)	125 (236)	131 (236)	95 (236)
114 (236)	90 (236)	93 (236)	127 (236)	139 (236)	95 (236)	89 (236)	118 (236)	95 (236)	139 (236)
149 (236)	128 (236)	110 (236)	117 (236)	110 (236)	109 (236)	100 (236)	119 (236)	119 (236)	155 (236)
87 (236)	63 (236)	104 (236)	121 (236)	146 (236)	135 (236)	141 (236)	101 (236)	125 (236)	128 (236)
117 (236)	119 (236)	134 (236)	111 (236)	111 (236)	46 (236)	17 (236)	8 (236)	8 (236)	41 (236)
79 (236)	42 (236)	46 (236)	52 (236)	42 (236)	23 (236)	69 (236)	36 (236)	37 (236)	27 (236)
58 (236)	47 (236)	52 (236)	47 (236)	49 (236)	18 (236)	34 (236)	36 (236)	37 (236)	50 (236)
39 (236)	85 (236)	23 (236)	25 (236)	21 (236)	21 (236)	94 (236)	7 (236)	138 (236)	29 (236)
51 (236)	88 (236)	109 (236)	8 (237)	4 (238)	16 (239)	8 (239)	11 (239)	7 (239)	22 (239)
1 (400)	8 (404)	1 (404)	1 (404)	3 (404)	5 (404)	1 (410)	3 (410)	2 (410)	1 (410)
1 (410)	8 (410)	2 (410)	3 (410)	1 (410)	1 (410)	2 (413)	1 (413)	1 (413)	17 (413)
13 (415)	1 (415)	10 (415)	3 (415)	1 (417)	7 (417)	0 (417)	1 (420)	2 (638)	1546 (638)
8 (648)	1 (648)	17 (643)							

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 333
MEAN = 56
MAX = 1546
MIN = 0

SCLEROSCOPIC HARDNESS

74 (3)	90 (3)	71 (5)	75 (7)	69 (7)	84 (7)	77 (7)	53 (7)	71 (7)	49 (7)
85 (7)	71 (7)	76 (7)	77 (7)	69 (7)	100 (11)	92 (13)	76 (13)	70 (13)	81 (13)
80 (13)	84 (13)	84 (13)	64 (13)	63 (13)	96 (13)	72 (13)	90 (14)	95 (14)	84 (14)
82 (14)	83 (14)	77 (14)	67 (14)	63 (14)	71 (14)	70 (14)	66 (14)	82 (21)	40 (21)
83 (21)	66 (21)	80 (21)	76 (21)	90 (22)	88 (22)	95 (22)	95 (22)	95 (22)	98 (22)
100 (22)	44 (22)	85 (22)	80 (22)	85 (22)	89 (22)	85 (22)	37 (22)	59 (22)	53 (22)
92 (22)	94 (22)	104 (22)	97 (22)	99 (22)	99 (22)	95 (22)	91 (22)	85 (22)	97 (22)
85 (22)	100 (22)	81 (22)	97 (22)	75 (22)	100 (22)	86 (22)	83 (22)	99 (23)	65 (34)
87 (39)	66 (45)	79 (45)	93 (45)	69 (45)	88 (45)	75 (45)	70 (47)	60 (47)	10 (47)
34 (47)	49 (47)	36 (47)	69 (48)	64 (48)	64 (48)	69 (48)	77 (48)	77 (48)	77 (48)
86 (48)	78 (50)	90 (52)	82 (52)	70 (52)	83 (59)	80 (64)	54 (64)	57 (65)	58 (65)
57 (65)	51 (65)	73 (65)	73 (65)	19 (201)	24 (201)	7 (201)	101 (208)	84 (208)	104 (208)
27 (208)	67 (211)	46 (211)	74 (211)	98 (211)	30 (212)	17 (218)	74 (218)	66 (218)	69 (218)
71 (218)	42 (218)	61 (218)	53 (218)	58 (218)	39 (218)	36 (218)	52 (218)	27 (223)	27 (223)
58 (223)	52 (223)	61 (223)	66 (223)	54 (223)	63 (223)	56 (223)	64 (223)	52 (223)	61 (223)
58 (223)	33 (223)	52 (223)	13 (223)	30 (223)	36 (223)	55 (223)	46 (223)	10 (223)	13 (223)
16 (223)	16 (223)	8 (223)	52 (223)	48 (223)	51 (223)	33 (223)	48 (223)	54 (223)	59 (223)
45 (223)	48 (223)	54 (223)	54 (223)	59 (223)	65 (223)	48 (223)	41 (223)	54 (223)	52 (223)
46 (223)	41 (223)	56 (223)	46 (223)	34 (223)	53 (223)	52 (223)	52 (223)	3 (223)	38 (223)
52 (223)	14 (223)	54 (223)	13 (223)	42 (223)	42 (223)	52 (223)	52 (223)	14 (223)	54 (223)
13 (223)	58 (223)	56 (223)	57 (223)	64 (223)	44 (223)	66 (223)	58 (223)	43 (223)	37 (223)
28 (223)	45 (223)	46 (223)	48 (223)	47 (223)	60 (223)	64 (223)	36 (223)	41 (223)	40 (223)
14 (223)	23 (223)	62 (225)	47 (225)	56 (225)	47 (225)	59 (225)	47 (225)	57 (225)	44 (225)
46 (225)	51 (225)	49 (225)	55 (225)	61 (225)	23 (225)	59 (228)	31 (236)	20 (236)	31 (236)
20 (236)	58 (236)	56 (236)	53 (236)	63 (236)	62 (236)	50 (236)	23 (236)	33 (236)	50 (236)
55 (236)	44 (236)	65 (236)	29 (236)	36 (236)	42 (236)	54 (236)	32 (236)	21 (236)	21 (236)
25 (236)	21 (236)	25 (236)	21 (236)	25 (236)	55 (236)	42 (236)	56 (236)	26 (236)	42 (236)
24 (236)	34 (236)	61 (236)	26 (236)	34 (236)	26 (236)	28 (236)	31 (236)	51 (236)	43 (236)
39 (236)	57 (236)	73 (236)	61 (236)	42 (236)	86 (236)	22 (236)	51 (236)	50 (236)	28 (236)
14 (236)	45 (237)	65 (237)	34 (237)	71 (237)	58 (237)	51 (237)	62 (237)	36 (237)	38 (237)
42 (237)	37 (237)	31 (237)	48 (237)	42 (237)	44 (237)	50 (237)	13 (238)	53 (238)	71 (238)
73 (238)	49 (238)	41 (238)	20 (238)	19 (238)	29 (238)	12 (248)	95 (257)	94 (257)	95 (257)
92 (400)	84 (400)	86 (400)	93 (400)	64 (400)	84 (400)	78 (400)	81 (400)	97 (404)	99 (404)
97 (404)	99 (404)	75 (404)	81 (404)	85 (404)	89 (404)	75 (404)	71 (404)	82 (404)	74 (404)
74 (404)	80 (404)	87 (404)	99 (404)	71 (404)	96 (404)	81 (407)	31 (407)	71 (407)	40 (407)
37 (407)	49 (407)	80 (407)	78 (407)	64 (407)	60 (407)	56 (410)	56 (410)	49 (410)	46 (410)
43 (410)	39 (410)	46 (410)	46 (410)	38 (410)	66 (410)	55 (410)	42 (410)	43 (410)	52 (410)
41 (410)	49 (410)	40 (412)	81 (413)	55 (413)	83 (413)	73 (413)	73 (413)	77 (413)	34 (413)
74 (413)	71 (413)	97 (413)	94 (413)	78 (413)	74 (413)	82 (414)	71 (415)	61 (416)	79 (416)
56 (417)	46 (417)	75 (417)	59 (417)	59 (417)	72 (419)	65 (419)	78 (419)	47 (422)	68 (422)
56 (423)	42 (423)	31 (439)	42 (439)	46 (440)	85 (440)	84 (440)	85 (440)	22 (607)	6 (607)
6 (607)	85 (622)	63 (622)	96 (622)	74 (622)	51 (622)	49 (622)	50 (622)	53 (622)	58 (622)
52 (622)	57 (622)	44 (622)	50 (622)	72 (629)	60 (629)	53 (629)	51 (629)	89 (638)	18 (638)
41 (648)	42 (648)	46 (648)							

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N =	433
MEAN =	59
MAX =	104
MIN =	6

```
MH33060.   03/17/70.PURDUE MACE 02/07/70.

11.13.22.MH330/     15073,PERLOFF,T180,CM65000,L30
11.13.22.00,TP1,P10.
11.13.22.RUN(S)
11.47.19.CTIME 001.395 SEC.  RUN MOD LEVEL 48
11.47.20.REQUEST(TAPE1,556,HY,A,C=R4,MT,READ)
02.08.31.  MT51 ASSIGNED = 556
02.08.32.REWIND(TAPE1)
02.08.32.LGO.
02.08.34.CX     1.533 SEC.
02.08.34.PX     5.411 SEC.
02.08.34.NL    43000
02.32.54.END     MAIN
02.32.54.CP    47.039 SEC.
02.32.54.PP   340.884 SEC.
02.32.54.LINES =          1521 OCTAL
02.32.54.CM     3.041 MWD-SEC.
```

Program No. 4

Program to print out a particular rock type of all values within specified ranges for a specific selection of variables

```
000000          PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
000003          DATA B1,B2,B3,B4/8400,97,2026,48/
000003          DATA X/1H /
000003          DIMENSION CARD(8)
000003          DIMENSION IRT(2171),DUMMY(2), PERM(1000),SGT(700),SGA(1600),SGU(30
                1),PORO(1300),ABSOP(500),SCLERO(500),ABRAS(200),TOUGH(300),RT1(100
                20),RT2(700),RT3(1600),RT4(300),RT5(1300),RT6(500),RT7(500),RT8(200
                3),RT9(300)
000003          INTEGER RT1,RT2,RT3,RT4,RT5,RT6,RT7,RT8,RT9,DUMMY
000003          INTEGER A2,A3
000003          REAL MEAN,MAX,MIN
000003          EQUIVALENCE (IRT(1),DUMMY(2))
000003          DUMMY(1) = 3
000004          N1 = 0
000005          N2 = 0
000006          N3 = 0
000007          N4 = 0
000010          N5 = 0
000011          N6 = 0
000012          N7 = 0
000013          N8 = 0
000014          N9 = 0
000015          DO 100 K=1,2171
000016          READ(1,400) CARD
000033      400 FORMAT(8A10)
000033          DECODE(80,2,CARD(1)) IRT(K),A1,A2,A3,A4,A5,A6,A7,A8,A9
000065        2 FORMAT(15X,I3,14X,A1,3X,I4,2X,I4,2X,A4,2X,A3,2X,A4,2X,A3,2X,A2,2X,
                1A3)
000065          IF(EOF,1) 99,101
000040      101 READ(1,3) B
000066          READ(1,3) C
000074          READ(1,3) D
000102          READ(1,3) E
000110          READ(1,3) F
000116          READ(1,3) G
000124          READ(1,3) H
000132        3 FORMAT(50X,A5)
000132          IF((IRT(K).NE.IRT(K-1))) GO TO 50
000135       11 IF(A1.NE.X) GO TO 31
000137       12 IF((A2.LT.6400).AND.(A2.GT.97)) GO TO 32
000147       13 IF((A3.LT.2026).AND.(A3.GT.48)) GO TO 33
000157       14 IF(A4.NE.X) GO TO 34
000161       15 IF(A5.NE.X) GO TO 35
000163       16 IF(A6.NE.X) GO TO 36
000165       17 IF(A7.NE.X) GO TO 37
000167       18 IF(A8.NE.X) GO TO 38
000171       19 IF(A9.NE.X) GO TO 39
000173          GO TO 100
000174       31 N1 = N1 + 1
000176          DECODE(80,40,CARD(1)) PERM(N1)
000205       40 FORMAT(32X,F1.0)
000205          RT1(N1) = IRT(K)
000210          GO TO 12
000210       32 N2 = N2 + 1
000212          DECODE(80,41,CARD(1)) SGT(N2)
000231       41 FORMAT(36X,F4.0)
000231          RT2(N2) = IRT(K)
000234          GO TO 13
000234       33 N3 = N3 + 1
000236          DECODE(80,42,CARD(1)) SGA(N3)
000235       42 FORMAT(42X,F4.0)
000235          RT3(N3) = IRT(K)
000240          GO TO 14
000240       34 N4 = N4 + 1
000242          DECODE(80,43,CARD(1)) SGU(N4)
000251       43 FORMAT(48X,F4.0)
000251          RT4(N4) = IRT(K)
000254          GO TO 15
000254       35 N5 = N5 + 1
000256          DECODE(80,44,CARD(1)) PORO(N5)
000265       44 FORMAT(54X,F3.0)
000265          RT5(N5) = IRT(K)
000270          GO TO 14
000270       36 N6 = N6 + 1
000272          DECODE(80,45,CARD(1)) ABSOP(N6)
000301       45 FORMAT(59X,F4.0)
000301          RT6(N6) = IRT(K)
000304          GO TO 17
000304       37 N7 = N7 + 1
000306          DECODE(80,46,CARD(1)) SCLERO(N7)
000315       46 FORMAT(65X,F3.0)
000315          RT7(N7) = IRT(K)
000320          GO TO 18
000320       38 N8 = N8 + 1
000322          DECODE(80,47,CARD(1)) ABRAS(N8)
000331       47 FORMAT(70X,F2.0)
000331          RT8(N8) = IRT(K)
000334          GO TO 19
000334       39 N9 = N9 + 1
000336          DECODE(80,48,CARD(1)) TOUGH(N9)
000345       48 FORMAT(74X,F3.0)
000345          RT9(N9) = IRT(K)
000350          GO TO 100
000350       50 IF(N1.EQ.0) GO TO 501
000351          WRITE(6,51)
000355       51 FORMAT(1H1,40X,12HPERMEABILITY//)
000355          WRITE(6,60) (PERM(I),RT1(I),I=1,N1)
000372       60 FORMAT(10(F6.0,2H ,,I3,1H))
```

```
000372          WRITE(6,70)
000376       70 FORMAT(//10X,44H(NUMBERS IN PARENTHESES REFER TO ROCK TYPES))
000376          CALL MMM(N1,PERM,MEAN,MAX,MIN)
000422          WRITE(6,61) N1,MEAN,MAX,MIN
000474       61 FORMAT(///5X,3HN =,I6/2X,6HMEAN =,F6.0/3X,5HMAX =,F6.0/3X,5HMIN =,
                1F6.0)
000516      501 IF(N2.EQ.0) GO TO 502
000517          WRITE(6,52)
000523       52 FORMAT(1H0,40X,21HTRUE SPECIFIC GRAVITY//)
000523          WRITE(6,60) (SGT(I),RT2(I),I=1,N2)
000440          WRITE(6,70)
000444          CALL MMM(N2,SGT,MEAN,MAX,MIN)
000450          WRITE(6,61) N2,MEAN,MAX,MIN
000444      502 IF(N3.EQ.0) GO TO 503
000444          WRITE(6,53)
000471       53 FORMAT(1H0,40X,25HAPPARENT SPECIFIC GRAVITY//)
000471          WRITE(6,60) (SGA(I),RT3(I),I=1,N3)
000506          WRITE(6,70)
000512          CALL MMM(N3,SGA,MEAN,MAX,MIN)
000516          WRITE(6,61) N3,MEAN,MAX,MIN
000532      503 IF(N4.EQ.0) GO TO 504
000533          WRITE(6,54)
000537       54 FORMAT(1H0,40X,36HUNSPECIFIED TYPE OF SPECIFIC GRAVITY//)
000537          WRITE(6,60) (SGU(I),RT4(I),I=1,N4)
000564          WRITE(6,70)
000560          CALL MMM(N4,SGU,MEAN,MAX,MIN)
000564          WRITE(6,61) N4,MEAN,MAX,MIN
000600      504 IF(N5.EQ.0) GO TO 505
000601          WRITE(6,55)
000605       55 FORMAT(1H0,40X,8HPOROSITY//)
000605          WRITE(6,60) (PORO(I),RT5(I),I=1,N5)
000622          WRITE(6,70)
000626          CALL MMM(N5,PORO,MEAN,MAX,MIN)
000632          WRITE(6,61) N5,MEAN,MAX,MIN
000646      505 IF(N6.EQ.0) GO TO 506
000647          WRITE(6,56)
000653       56 FORMAT(1H0,40X,10HABSORPTION//)
000653          WRITE(6,60) (ABSOP(I),RT6(I),I=1,N6)
000670          WRITE(6,70)
000674          CALL MMM(N6,ABSOP,MEAN,MAX,MIN)
000700          WRITE(6,61) N6,MEAN,MAX,MIN
000714      506 IF(N7.EQ.0) GO TO 507
000715          WRITE(6,57)
000731       57 FORMAT(1H0,40X,21HSCLEROSCOPIC HARDNESS//)
000731          WRITE(6,60) (SCLERO(I),RT7(I),I=1,N7)
000736          WRITE(6,70)
000742          CALL MMM(N7,SCLERO,MEAN,MAX,MIN)
000746          WRITE(6,61) N7,MEAN,MAX,MIN
000762      507 IF(N8.EQ.0) GO TO 508
000763          WRITE(6,58)
000767       58 FORMAT(1H0,40X,17HABRASIVE HARDNESS//)
000767          WRITE(6,60) (ABRAS(I),RT8(I),I=1,N8)
001004          WRITE(6,70)
001010          CALL MMM(N8,ABRAS,MEAN,MAX,MIN)
001014          WRITE(6,61) N8,MEAN,MAX,MIN
001030      508 IF(N9.EQ.0) GO TO 88
001031          WRITE(6,59)
001035       59 FORMAT(1H0,40X,16HIMPACT TOUGHNESS//)
001035          WRITE(6,60) (TOUGH(I),RT9(I),I=1,N9)
001062          WRITE(6,70)
001066          CALL MMM(N9,TOUGH,MEAN,MAX,MIN)
001062          WRITE(6,61) N9,MEAN,MAX,MIN

001076       88 DO 81 J=1,N1
001100          PERM(J) = 0.0
001101       81 RT1(J) = 0.0
001104          DO 82 J=1,N2
001106          SGT(J) = 0.0
001107       82 RT2(J) = 0.0
001112          DO 83 J=1,N3
001114          SGA(J) = 0.0
001115       83 RT3(J) = 0.0
001120          DO 84 J=1,N4
001122          SGU(J) = 0.0
001123       84 RT4(J) = 0.0
001126          DO 85 J=1,N5
001130          PORO(J) = 0.0
001131       85 RT5(J) = 0.0
001134          DO 86 J=1,N6
001136          ABSOP(J) = 0.0
001137       86 RT6(J) = 0.0
001142          DO 87 J=1,N7
001144          SCLERO(J) = 0.0
001145       87 RT7(J) = 0.0
001150          DO 88 J=1,N8
001152          ABRAS(J) = 0.0
001153       88 RT8(J) = 0.0
001156          DO 89 J=1,N9
001160          TOUGH(J) = 0.0
001161       89 RT9(J) = 0.0
001164          N1 = 0
001165          N2 = 0
001166          N3 = 0
001167          N4 = 0
001170          N5 = 0
001171          N6 = 0
001172          N7 = 0
001173          N8 = 0
001174          N9 = 0
001174          GO TO 11
001175      100 CONTINUE
001177       99 IF(N1.EQ.0) GO TO 202
001200          WRITE(6,51)
001204          WRITE(6,60) (PERM(I),RT1(I),I=1,N1)
001221          WRITE(6,70)
001225          CALL MMM(N1,PERM,MEAN,MAX,MIN)
001231          WRITE(6,61) N1,MEAN,MAX,MIN
001245      202 IF(N2.EQ.0) GO TO 203
001246          WRITE(6,52)
001252          WRITE(6,60) (SGT(I),RT2(I),I=1,N2)
001267          WRITE(6,70)
001273          CALL MMM(N2,SGT,MEAN,MAX,MIN)
001277          WRITE(6,61) N2,MEAN,MAX,MIN
001313      203 IF(N3.EQ.0) GO TO 204
001314          WRITE(6,53)
001320          WRITE(6,60) (SGA(I),RT3(I),I=1,N3)
001335          WRITE(6,70)
001341          CALL MMM(N3,SGA,MEAN,MAX,MIN)
001345          WRITE(6,61) N3,MEAN,MAX,MIN
001361      204 IF(N4.EQ.0) GO TO 205
001362          WRITE(6,54)
001366          WRITE(6,60) (SGU(I),RT4(I),I=1,N4)
001403          WRITE(6,70)
001407          CALL MMM(N4,SGU,MEAN,MAX,MIN)
001413          WRITE(6,61) N4,MEAN,MAX,MIN
```

```
001427      205 IF(N5.EQ.0) GO TO 206
001430          WRITE(6,55)
001434          WRITE(6,60) (PORO(I),RT5(I),I=1,N5)
001451          WRITE(6,70)
001455          CALL MMM(N5,PORO,MEAN,MAX,MIN)
001461          WRITE(6,61) N5,MEAN,MAX,MIN
001475      206 IF(N6.EQ.0) GO TO 207
001476          WRITE(6,56)
001502          WRITE(6,60) (ABSOP(I),RT6(I),I=1,N6)
001517          WRITE(6,70)
001523          CALL MMM(N6,ABSOP,MEAN,MAX,MIN)
001527          WRITE(6,61) N6,MEAN,MAX,MIN
001543      207 IF(N7.EQ.0) GO TO 208
001544          WRITE(6,57)
001550          WRITE(6,60) (SCLERO(I),RT7(I),I=1,N7)
001565          WRITE(6,70)
001571          CALL MMM(N7,SCLERO,MEAN,MAX,MIN)
001575          WRITE(6,61) N7,MEAN,MAX,MIN
001611      208 IF(N8.EQ.0) GO TO 209
001612          WRITE(6,58)
001616          WRITE(6,60) (ABRAS(I),RT8(I),I=1,N8)
001633          WRITE(6,70)
001637          CALL MMM(N8,ABRAS,MEAN,MAX,MIN)
001643          WRITE(6,61) N8,MEAN,MAX,MIN
001657      209 IF(N9.EQ.0) GO TO 210
001660          WRITE(6,59)
001664          WRITE(6,60) (TOUGH(I),RT9(I),I=1,N9)
001701          WRITE(6,70)
001705          CALL MMM(N9,TOUGH,MEAN,MAX,MIN)
001711          WRITE(6,61) N9,MEAN,MAX,MIN
001725      210 STOP
001727          END

PROGRAM LENGTH INCLUDING I/O BUFFERS
005266

UNUSED COMPILER SPACE
015100


                SUBROUTINE MMM(N,X,MEAN,MAX,MIN)
000010          DIMENSION X(2000)
000010          REAL MEAN,MAX,MIN
000010          SUM = 0.0
000010          MAX = 0.0
000011          MIN = 10000.0
000012          DO 100 I=1,N
000014          SUM = SUM + X(I)
000016          IF(X(I).GE.MAX) MAX = X(I)
000023      100 IF(X(I).LE.MIN) MIN = X(I)
000032          MEAN = SUM/FLOAT(N)
000034          RETURN
000036          END

SUBPROGRAM LENGTH
000051

UNUSED COMPILER SPACE
022500
```

LOAD MAP FILE = LGO 23.17.52. 03/16/70. PAGE 1

FWA LOAD 100 LWA LOAD 47760 FWA LOADER 57277 FWA TABLES 56513 UNUSED STORAGE 4533

PROGRAM	ADDRESS	FILE	COMMON	ADDRESS	LENGTH
MAIN	100	LGO			
MMM	42566	LGO			
IFENDF	42637	SYSTEM			
INPUTC	42671	SYSTEM			
SYSTEM	44012	SYSTEM	SCOPE2	44012	
INPUTS	45063	SYSTEM			
OUTPTC	45145	SYSTEM			
FATAL78	46505	SYSTEM			
SIOS	46623	SYSTEM			
GETBA	47741	SYSTEM			
			/BLANK/	0	0

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
MAIN	101								
MMM	42567	MAIN	401	447	515	563	631	677	745	1013
			1061	1230	1276	1344	1412	1460	1526	1574
			1642	1710						
IFENDF	42640	MAIN	56							
INPUTC	42873	MAIN	17	21	22	62	64	65	70	72
			73	76	100	101	104	106	107	112
			114	115	120	122	123	126	130	131
KRAKER	42775	INPUTS	5	23						
QENTRY	44013	MAIN	2							
SYSTEM	44220	IFENDF	22							
		INPUTC	37	727						
		INPUTS	26							
		OUTPTC	14	1144						
SYSTEMC	44164									
SYSTEMP	44213									
END	44107	MAIN	1730							
		MMM	35							
STOP	44137	MAIN	1726							
EXIT	44131									
ABNORML	44147	IFENDF	23							
		INPUTC	40	730						
		INPUTS	27							
		OUTPTC	15	1145						
INPUTS	45065	MAIN	25	27	31	33	35	37	41	43
			45	47	51	53	54	177	201	203
			204	213	215	217	220	227	231	233
			234	243	245	247	250	257	261	263
			264	273	275	277	300	307	311	313
			314	323	325	327	330	337	341	343
			344							
OUTPTC	45147	MAIN	353	354	357	363	365	371	374	375
			404	406	410	412	414	415	421	422
			425	431	433	437	442	443	452	454
			456	460	462	463	467	470	473	477
			501	505	510	511	520	522	524	526
			530	531	535	536	541	545	547	553

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
		556	557	566	570	572	574	576	577
		603	604	607	613	615	621	624	625
		634	636	640	642	644	645	651	652
		655	661	663	667	672	673	702	704
		706	710	712	713	717	720	723	727
		731	735	740	741	750	752	754	758
		760	761	765	766	771	775	777	1003
		1006	1007	1016	1020	1022	1024	1026	1027
		1033	1034	1037	1043	1045	1051	1054	1055
		1064	1066	1070	1072	1074	1075	1202	1203
		1206	1212	1214	1220	1223	1224	1233	1235
		1237	1241	1243	1244	1250	1251	1254	1260
		1262	1266	1271	1272	1301	1303	1305	1307
		1311	1312	1316	1317	1322	1326	1330	1334
		1337	1340	1347	1351	1353	1355	1357	1360
		1364	1365	1370	1374	1376	1402	1405	1406
		1415	1417	1421	1423	1425	1426	1432	1433
		1436	1442	1444	1450	1453	1454	1463	1465
		1467	1471	1473	1474	1500	1501	1504	1510
		1512	1516	1521	1522	1631	1533	1535	1537
		1541	1542	1546	1547	1552	1556	1560	1564
		1567	1570	1577	1601	1603	1605	1607	1610
		1614	1615	1620	1624	1626	1632	1636	1636
		1645	1647	1651	1653	1655	1656	1662	1663
		1666	1672	1674	1700	1703	1704	1713	1715
		1717	1721	1723	1724				
KODER	45306								
FATAL7R	45506	SYSTEM	710						
CARD.F	46603	SIOs	56						
BKSPRU.	47112								
FIZBAK.	47122								
POSFIL.	47150	OUTPTC	32						
RDPRIU.	47240								
DAT.	47261	OUTPTC	56	72	37				
CIOI.	47066								
OPEN.	46623	INPUTC	22						
		SYSTEM	506						
		OUTPTC	22						
SIO.	44744	INPUTC	52						
		OUTPTC	71						
ADVIN.	47161	SYSTEM	410						
MYWDS.	44773								
POSFI.	47171								
FIZBA.	47203								
DAT..	47511	INPUTC	56	24	47				
		INPUTS	17	33	42	45			
BETBA	47741	IFENDF	4						
		INPUTC	10						
		OUTPTC	10						

UNSATISFIED EXTERNALS REFERENCES (RELATIVE)

** NONE **

PERMEABILITY

2 (45)	3 (45)	2 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	1 (45)	2 (45)
2 (45)	1 (45)	2 (45)	2 (45)	2 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)
1 (45)	1 (45)	1 (45)	2 (45)	2 (45)	2 (45)	2 (45)	2 (45)	1 (45)	1 (45)
1 (45)	1 (45)	1 (45)	2 (45)	1 (45)	2 (45)	1 (45)	1 (45)	2 (45)	1 (45)
3 (45)	3 (45)	1 (45)	2 (45)	2 (45)	1 (45)	1 (45)	2 (45)	2 (45)	2 (45)
1 (45)	2 (45)	1 (45)	2 (45)	1 (45)	3 (45)	3 (45)	2 (45)	2 (45)	2 (45)
2 (45)	2 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)	2 (45)	1 (45)	1 (45)
1 (45)	1 (45)	2 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)
1 (45)	1 (45)	2 (45)	1 (45)	2 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)
1 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)	1 (45)	2 (45)
2 (45)	1 (45)								

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 102
MEAN = 1
MAX = 3
MIN = 1

TRUE SPECIFIC GRAVITY

217 (45)	247 (45)	252 (45)	252 (45)	242 (45)	246 (45)	256 (45)	255 (45)	242 (45)	250 (45)
261 (45)	259 (45)	254 (45)	254 (45)	254 (45)	252 (45)	250 (45)	252 (45)	256 (45)	243 (45)
251 (45)	251 (45)	250 (45)	259 (45)	257 (45)	256 (45)	254 (45)	253 (45)	255 (45)	254 (45)
263 (45)	263 (45)	263 (45)	261 (45)	261 (45)	261 (45)	250 (45)	259 (45)	260 (45)	262 (45)
251 (45)	248 (45)	248 (45)	259 (45)	249 (45)	250 (45)	250 (45)	260 (45)	260 (45)	249 (45)
252 (45)	256 (45)	250 (45)	257 (45)	250 (45)	257 (45)	257 (45)	256 (45)	256 (45)	254 (45)
261 (45)	261 (45)	259 (45)	259 (45)	251 (45)	258 (45)	259 (45)	260 (45)	261 (45)	261 (45)
261 (45)	262 (45)	260 (45)	256 (45)	255 (45)	258 (45)	258 (45)	260 (45)	260 (45)	260 (45)
256 (45)	254 (45)	261 (45)	258 (45)	257 (45)	261 (45)	259 (45)	257 (45)	256 (45)	263 (45)
263 (45)	263 (45)	256 (45)	261 (45)	251 (45)	261 (45)	258 (45)	257 (45)	255 (45)	255 (45)
257 (45)	267 (45)	271 (45)	281 (45)	287 (45)	281 (45)				

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 106
MEAN = 257
MAX = 287
MIN = 217

APPARENT SPECIFIC GRAVITY

270 (45)	194 (45)	135 (45)	211 (45)	229 (45)	209 (45)	228 (45)	212 (45)	217 (45)	229 (45)
201 (45)	232 (45)	239 (45)	225 (45)	226 (45)	222 (45)	230 (45)	225 (45)	231 (45)	240 (45)
223 (45)	231 (45)	236 (45)	229 (45)	215 (45)	226 (45)	224 (45)	226 (45)	224 (45)	242 (45)
244 (45)	239 (45)	239 (45)	239 (45)	209 (45)	243 (45)	228 (45)	230 (45)	235 (45)	233 (45)
248 (45)	176 (45)	184 (45)	243 (45)	224 (45)	217 (45)	230 (45)	230 (45)	228 (45)	224 (45)
229 (45)	232 (45)	226 (45)	236 (45)	218 (45)	234 (45)	197 (45)	197 (45)	221 (45)	221 (45)
221 (45)	231 (45)	231 (45)	234 (45)	234 (45)	234 (45)	237 (45)	243 (45)	214 (45)	246 (45)
246 (45)	246 (45)	238 (45)	219 (45)	248 (45)	243 (45)	252 (45)	246 (45)	249 (45)	247 (45)
240 (45)	245 (45)	240 (45)	226 (45)	235 (45)	219 (45)	235 (45)	245 (45)	246 (45)	248 (45)
256 (45)	255 (45)	255 (45)	239 (45)	237 (45)	237 (45)	237 (45)	238 (45)	233 (45)	223 (45)
223 (45)	228 (45)	228 (45)	315 (45)	228 (45)	255 (45)	247 (45)	234 (45)	254 (45)	262 (45)
262 (45)	265 (45)	269 (45)	279 (45)	284 (45)	276 (45)				

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N =	118
MEAN =	235
MAX =	315
MIN =	135

N.B. These refer to Apparent Specific Gravity on preceding page.

UNSPECIFIED TYPE OF SPECIFIC GRAVITY

252 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N =	1
MEAN =	252
MAX =	252
MIN =	252

POROSITY

104 (45)	455 (45)	163 (45)	93 (45)	137 (45)	115 (45)	172 (45)	149 (45)	55 (45)	107 (45)
100 (45)	71 (45)	112 (45)	108 (45)	126 (45)	89 (45)	100 (45)	81 (45)	101 (45)	93 (45)
91 (45)	59 (45)	83 (45)	164 (45)	118 (45)	124 (45)	116 (45)	113 (45)	53 (45)	41 (45)
93 (45)	93 (45)	93 (45)	198 (45)	67 (45)	126 (45)	80 (45)	93 (45)	104 (45)	51 (45)
290 (45)	257 (45)	20 (45)	134 (45)	130 (45)	80 (45)	80 (45)	122 (45)	122 (45)	122 (45)
78 (45)	116 (45)	56 (45)	149 (45)	98 (45)	234 (45)	234 (45)	138 (45)	138 (45)	138 (45)
113 (45)	113 (45)	97 (45)	97 (45)	98 (45)	85 (45)	61 (45)	161 (45)	60 (45)	40 (45)
60 (45)	92 (45)	157 (45)	33 (45)	48 (45)	23 (45)	47 (45)	42 (45)	42 (45)	42 (45)
37 (45)	55 (45)	134 (45)	90 (45)	146 (45)	99 (45)	53 (45)	44 (45)	32 (45)	30 (45)
30 (45)	30 (45)	70 (45)	90 (45)	90 (45)	90 (45)	77 (45)	92 (45)	128 (45)	158 (45)
102 (45)	61 (45)	9 (45)	25 (45)	31 (45)	27 (45)	7 (45)	7 (45)	7 (45)	10 (45)
18 (45)									

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N =	111
MEAN =	96
MAX =	455
MIN =	7

ABSORPTION

2 (45)	8 (45)	2 (45)	1 (45)	3 (45)	3 (45)	3 (45)	4 (45)	7 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N =	9
MEAN =	4
MAX =	8
MIN =	1

SCLEROSCOPIC HARDNESS

66 (45) 79 (45) 93 (45) 69 (45) 88 (45) 75 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 6
MEAN = 78
MAX = 93
MIN = 66

IMPACT TOUGHNESS

62 (45)

(NUMBERS IN PARENTHESES REFER TO ROCK TYPES)

N = 1
MEAN = 62
MAX = 62
MIN = 62

MH32916. 03/17/70.PURDUE MACE 02/07/70.

11.49.16.MH329/ 15073,PERLOFF,T60,CM60000 0
11.49.16.000,TP1,P10.
11.49.16.MAP(ON)
11.49.16.RUN(S)
11.51.12.CTIME 002.683 SEC. RUN MOD LEVEL 4B
11.51.13.REQUEST(TAPE1,556,HY,X,C=84,MT,READ)
23.17.49. MT53 ASSIGNED = 556
23.17.49.REWIND(TAPE1)
23.17.50.LGO(LC=100000)
23.17.53.CX 3.224 SEC.
23.17.53.PX 5.396 SEC.
23.17.53.NL 50100
00.13.30.STOP
00.13.30.CP 52.597 SEC.
00.13.30.PP 358.094 SEC.
00.13.30.LINES = 014565 OCTAL
00.13.30.CM 5.277 MOD-SEC.

Program No. 5

LEAST SQUARES ROUTINE

for non-blank entries for both variables
in the same data set for selected pairs
of variables

```
00003          PROGRAM LEAST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
00003          DATA X/1H /
00003          DIMENSION ACARD(8),BCARD(8),DCARD(8),FCARD(8),HCARD(8)
00003          DIMENSION IDEN(18),COEFF(20)
00003          DIMENSION B15A1(137),A1B15(137),B15D2(79),D2B15(79),B15F4(71),F4B1
               15(71),B15D9(110),D9B15(110),B15F10(61),F10B15(61),B15F11(59),F11B1
               25(59),B15A17(221),A17B15(221),B15A19(484),A19B15(484),B15A20(222),
               3A20B15(222),B15H21(28),H21B15(28),TYPE1(137),TYPE2(79),TYPE3(71),T
               4YPE4(110),TYPE5(61),TYPE6(59),TYPE7(221),TYPE8(484),TYPE9(222),TYP
               5F10(28),S1(137),S2(79),S3(71),S4(110),S5(61),S6(59),S7(221),S8(484
               6),S9(222),S10(28)
00003          INTEGER TYPE,TYPE1,TYPE2,TYPE3,TYPE4,TYPE5,TYPE6,TYPE7,TYPE8,TYPE9
               1,TYPE10
00003          DATA J1,J2,J3,J4,J5,J6,J7,J8,J9,J10/10*0/
00003          DO 100 K=1,2170
00005          READ(1,400)ACARD
00012      400 FORMAT(8A10)
00012          DECODE(80,1,ACARD(1)) S,TYPE,A20,A19,A17,A1
00034      1   FORMAT(7X,A1,7X,I3,18X,A4,2X,A4,19X,A3,2X,A2)
00034          READ(1,400) BCARD
00042          DECODE(80,2,BCARD(1)) B15
00052      2   FORMAT(28X,A3)
00052          READ(1,3) C
00060      3   FORMAT(50X,A5)
00060          READ(1,400) DCARD
00066          DECODE(80,24,DCARD(1)) D2,D9
00100      24  FORMAT(54X,A4,A3)
00100          READ(1,3) E
00106          READ(1,400) FCARD
00114          DECODE(80,25,FCARD(1)) F10,F4,F11
00130      25  FORMAT(14X,A3,15X,A4,A3)
00130          READ(1,3) G
00136          READ(1,400) HCARD
00144          DECODE(80,26,HCARD(1)) H21
00154      26  FORMAT(21A4)
00154      10  IF((B15.NE.X).AND.( A1.NE.X)) GO TO 50
00164      11  IF((B15.NE.X).AND.(A17.NE.X)) GO TO 51
00174      12  IF((B15.NE.X).AND.(A19.NE.X)) GO TO 52
00204      13  IF((B15.NE.X).AND.(A20.NE.X)) GO TO 53
00214      14  IF((B15.NE.X).AND.( D2.NE.X)) GO TO 54
00224      15  IF((B15.NE.X).AND.( D9.NE.X)) GO TO 55
00234      16  IF((B15.NE.X).AND.(F10.NE.X)) GO TO 56
00244      17  IF((B15.NE.X).AND.(F4 .NE.X)) GO TO 57
00254      18  IF((B15.NE.X).AND.(F11.NE.X)) GO TO 58
00264      19  IF((B15.NE.X).AND.(H21.NE.X)) GO TO 59
00274          GO TO 100
00274      50  J1 = J1 + 1
00276          DECODE(80,40,BCARD(1)) B15A1(J1)
00305      40  FORMAT(28X,F3.1)
00305          DECODE(80,440,ACARD(1)) A1B15(J1)
00315      440 FORMAT(70X,F2.0)
00315          TYPE1(J1) = TYPE
00317          S1(J1) = S
00321          GO TO 11
00321      51  J2 = J2 + 1
00323          DECODE(80,40,BCARD(1)) B15A17(J2)
00332          DECODE(80,441,ACARD(1)) A17B15(J2)
00342      441 FORMAT(65X,F3.0)
00342          TYPE2(J2) = TYPE
00344          S2(J2) = S
00346          GO TO 12
00346      52  J3 = J3 + 1
000350         DECODE(80,40,BCARD(1)) B15A19(J3)
000357         DECODE(80,442,ACARD(1)) A19B15(J3)
000367     442 FORMAT(42X,F4.2)
000367         TYPE3(J3) = TYPE
000371         S3(J3) = S
000373         GO TO 13
000373     53  J4 = J4 + 1
000375         DECODE(80,40,BCARD(1)) B15A20(J4)
000404         DECODE(80,443,ACARD(1)) A20B15(J4)
000414     443 FORMAT(36X,F4.2)
000414         TYPE4(J4) = TYPE
000416         S4(J4) = S
000420         GO TO 14
000420     54  J5 = J5 + 1
000422         DECODE(80,40,BCARD(1)) B15D2(J5)
000431         DECODE(80,444,DCARD(1)) D2B15(J5)
000441     444 FORMAT(54X,F4.2)
000441         TYPE5(J5) = TYPE
000443         S5(J5) = S
000445         GO TO 15
000445     55  J6 = J6 + 1
000447         DECODE(80,40,BCARD(1)) B15D9(J6)
000456         DECODE(80,445,DCARD(1)) D9B15(J6)
000466     445 FORMAT(58X,F3.2)
000466         TYPE6(J6) = TYPE
000470         S6(J6) = S
000472         GO TO 16
000472     56  J7 = J7 + 1
000474         DECODE(80,40,BCARD(1)) B15F10(J7)
000503         DECODE(80,446,FCARD(1)) F10B15(J7)
000513     446 FORMAT(14X,F3.2)
000513         TYPE7(J7) = TYPE
000515         S7(J7) = S
000517         GO TO 17
000517     57  J8 = J8 + 1
000521         DECODE(80,40,BCARD(1)) B15F4(J8)
000530         DECODE(80,447,FCARD(1)) F4B15(J8)
000540     447 FORMAT(32X,F4.2)
000540         TYPE8(J8) = TYPE
000542         S8(J8) = S
000544         GO TO 18
000544     58  J9 = J9 + 1
000546         DECODE(80,40,BCARD(1)) B15F11(J9)
000555         DECODE(80,448,FCARD(1)) F11B15(J9)
000565     448 FORMAT(36X,F3.2)
000565         TYPE9(J9) = TYPE
000567         S9(J9) = S
000571         GO TO 19
000571     59  J10 = J10 + 1
000573         DECODE(80,40,BCARD(1)) B15H21(J10)
000602         DECODE(80,449,HCARD(1)) H21B15(J10)
000612     449 FORMAT(21X,F4.2)
000612         TYPE10(J10) = TYPE
000614         S10(J10) = S
000616     100 CONTINUE
000620         NORDR = 1
000621         NRDR = NORDR + 1
000623         READ(5,20) IDEN
000631         WRITE(6,4) J1
```

```
000637          WRITE(6,5) (S1(I),TYPE1(I),B15A1(I),A1B15(I),I=1,J1)
000660          CALL LSTSQ(J1,NORDR,B15A1,A1B15,COEFF)
000664          WRITE(6,6) IDEN
000672          WRITE(6,7) (COEFF(I),I=1,NRDR)
000705          READ(5,20) IDEN
000713          WRITE(6,4) J2
000721          WRITE(6,5) (S2(I),TYPE2(I),B15A17(I),A17B15(I),I=1,J2)
000742          CALL LSTSQ(J2,NORDR,B15A17,A17B15,COEFF)
000746          WRITE(6,6) IDEN
000754          WRITE(6,7) (COEFF(I),I=1,NRDR)
000767          READ(5,20) IDEN
000775          WRITE(6,4) J3
001003          WRITE(6,5) (S3(I),TYPE3(I),B15A19(I),A19B15(I),I=1,J3)
001024          CALL LSTSQ(J3,NORDR,B15A19,A19B15,COEFF)
001030          WRITE(6,6) IDEN
001036          WRITE(6,7) (COEFF(I),I=1,NRDR)
001051          READ(5,20) IDEN
001057          WRITE(6,4) J4
001065          WRITE(6,5) (S4(I),TYPE4(I),B15A20(I),A20B15(I),I=1,J4)
001106          CALL LSTSQ(J4,NORDR,B15A20,A20B15,COEFF)
001112          WRITE(6,6) IDEN
001120          WRITE(6,7)(COEFF(I),I=1,NRDR)
001133          READ(5,20) IDEN
001141          WRITE(6,4) J5
001147          WRITE(6,5) (S5(I),TYPE5(I),B15D2(I),D2B15(I),I=1,J5)
001170          CALL LSTSQ(J5,NORDR,B15D2,D2B15,COEFF)
001174          WRITE(6,6) IDEN
0012            WRITE(6,7) (COEFF(I),I=1,NRDR)
001215          READ(5,20) IDEN
001223          WRITE(6,4) J6
001231          WRITE(6,5) (S6(I),TYPE6(I),B15D9(I),D9B15(I),I=1,J6)
001252          CALL LSTSQ(J6,NORDR,B15D9,D9B15,COEFF)
001256          WRITE(6,6) IDEN
001264          WRITE(6,7) (COEFF(I),I=1,NRDR)
001277          READ(5,20) IDEN
001305          WRITE(6,4) J7
001313          WRITE(6,5) (S7(I),TYPE7(I),B15F10(I),F10B15(I),I=1,J7)
001334          CALL LSTSQ(J7,NORDR,B15F10,F10B15,COEFF)
001340          WRITE(6,6) IDEN
001346          WRITE(6,7) (COEFF(I),I=1,NRDR)
001361          READ(5,20) IDEN
001367          WRITE(6,4) J8
001375          WRITE(6,5) (S8(I),TYPE8(I),B15F4(I),F4B15(I),I=1,J8)
001416          CALL LSTSQ(J8,NORDR,B15F4,F4B15,COEFF)
001422          WRITE(6,6) IDEN
001430          WRITE(6,7) (COEFF(I),I=1,NRDR)
001443          READ(5,20) IDEN
001451          WRITE(6,4) J9
001457          WRITE(6,5) (S9(I),TYPE9(I),B15F11(I),F11B15(I),I=1,J9)
001500          CALL LSTSQ(J9,NORDR,B15F11,F11B15,COEFF)
001504          WRITE(6,6) IDEN
001512          WRITE(6,7) (COEFF(I),I=1,NRDR)
001525          READ(5,20) IDEN
001533          WRITE(6,4) J10
001541          WRITE(6,5) (S10(I),TYPE10(I),B15H21(I),H21B15(I),I=1,J10)
001562          CALL LSTSQ(J10,NORDR,B15H21,H21B15,COEFF)
001566          WRITE(6,6) IDEN
001574          WRITE(6,7) (COEFF(I),I=1,NRDR)
001607      20  FORMAT(18A4)
001607      4   FORMAT(1H1,///10X,39HSTRAIGHT LINE LEAST SQUARES FIT THROUGH,I4,1X
               1,12HPOINTS (X,Y)///)
```

```
001607      5   FORMAT(6(4X,A1,1H=,F3.1H(,F5.0,1H,,F5.0,1H)))
001607      6   FORMAT(1H0,18A4)
001607      7   FORMAT(1H0,5X,3HY =,F8.3,3H + ,F8.3,4H * X)
001607          STOP
001611          END

PROGRAM LENGTH INCLUDING I/O BUFFERS
020627

UNUSED COMPILER SPACE
022300
                SUBROUTINE LSTSQ(NUM,NORDR,X,Y,RETRN)
000010          INTEGER R,RR                                                    LSTSQ002
000010          REAL MATRX(20,20),YSUM(20),RETRN(20)                            LSTSQ003
000010          DIMENSION X(500),Y(500),SX(40)
C                                                                               LSTSQ005
C       .....................................................................LSTSQ006
C                                                                               LSTSQ007
C          SUBROUTINE LSTSQ                                                     LSTSQ008
C          DECKS USED                                                           LSTSQ009
C             LSTSQ                                                             LSTSQ010
C                                                                               LSTSQ011
C          PURPOSE                                                              LSTSQ012
C             LSTSQ COMPUTES AN N TH-ORDER POLYNOMIAL LEAST SQUARES             LSTSQ013
C             FIT THROUGH M POINTS.                                             LSTSQ014
C                                                                               LSTSQ015
C          USAGE                                                                LSTSQ016
C             CALL LSTSQ(NUM,NORDR,X,Y,RETRN)                                   LSTSQ017
C                                                                               LSTSQ018
C          DESCRIPTION OF PARAMETERS                                            LSTSQ019
C             NUM    - NUMBER OF POINTS USED.                                   LSTSQ020
C             NORDR  - ORDER OF THE DESIRED POLYNOMIAL FIT.                     LSTSQ021
C             X      - NUM-VECTOR OF X-VALUES.                                  LSTSQ022
C             Y      - NUM-VECTOR OF CORRESPONDING Y-VALUES.                    LSTSQ023
C             RETRN  - (NORDR+1)-VECTOR OF COEFFICIENTS OF DESIRED              LSTSQ024
C                      POLYNOMIAL ORDERED FROM COEFFICIENT OF CONSTANT          LSTSQ025
C                      TERM TO COEFFICIENT OF HIGHEST POWER TERM.               LSTSQ026
C                                                                               LSTSQ027
C          REMARKS                                                              LSTSQ028
C             THE CURRENT DIMENSION STATEMENT IS SUFFICIENT FOR NORDR .LE.LSTSQ029
C             19 AND NUM .LE. 200. IF THE USER WISHES TO CHANGE THE             LSTSQ030
C             DIMENSION STATEMENT TO SUIT HIS NEEDS, THE FOLLOWING FORM         LSTSQ031
C             SHOULD BE USED--                                                  LSTSQ032
C                                                                               LSTSQ033
C             TYPE STATEMENT                                                    LSTSQ034
C                REAL MATRX(NORDR+1,NORDR+1),YSUM(NORDR+1),RETRN(NORDR+1)       LSTSQ035
C             DIMENSION STATEMENT                                               LSTSQ036
C                DIMENSION X(NUM),Y(NUM),SX(2*NORDR)                            LSTSQ037
C                                                                               LSTSQ038
C          METHOD                                                               LSTSQ039
C             A STANDARD LINEAR LEAST-SQUARES FIT IS APPROXIMATED BY            LSTSQ040
C             CALCULATING THE COEFFICIENTS OF THE NORMAL EQUATIONS AND           LSTSQ041
C             THEN SOLVING THE NORMAL EQUATIONS USING THE GAUSS-                LSTSQ042
C             ELIMINATION PROCESS. ORTHOGONAL COEFFICIENTS ARE NOT USED.        LSTSQ043
C             FOR REFERENCE--                                                   LSTSQ044
C             MCCRACKEN, D.D. AND DORN, W.S., #NUMERICAL METHODS AND            LSTSQ045
C             FORTRAN PROGRAMMING#,JOHN WILEY + SONS(NEW YORK--1964),           LSTSQ046
C             PP. 231-243, PP.262-275.                                          LSTSQ047
C          PROGRAMMER--RICHARD F. PUK                                           LSTSQ048
C                                                                               LSTSQ049
C       .....................................................................LSTSQ050
C                                                                               LSTSQ051
```

```
000010          NRDR = NORDR+1                                          LSTSQ052
                C ZERO MATRICES                                         LSTSQ053
000012          DO 10 I=1,NRDR                                          LSTSQ054
000013          YSUM(I)=0.                                              LSTSQ055
000015          DO 10 J=1,NRDR                                          LSTSQ056
000016       10 MATRX(I,J)=0.                                           LSTSQ057
000027          RR = 2*NORDR                                            LSTSQ058
000030          DO 11 I=1,RR                                            LSTSQ059
000031       11 SX(I)=0.                                                LSTSQ060
                C CALCULATE SUMS                                        LSTSQ061
000035          DO 20 K=1,RR                                            LSTSQ062
000037          DO 20 I=1,NUM                                           LSTSQ063
000040       20 SX(K)=SX(K)+X(I)**K                                     LSTSQ064
000052          DO 25 K=1,NORDR                                         LSTSQ065
000054          DO 25I=1,NUM                                            LSTSQ066
000055       25 YSUM(K+1) = YSUM(K+1)+(X(I)**K)*Y(I)                    LSTSQ067
000071          DO 26I=1,NUM                                            LSTSQ068
000072       26 YSUM(1) = YSUM(1)+Y(I)                                  LSTSQ069
                C INSERT IN MATRIX (CHANGE FOR DIFFERENT NORMAL EQUATIONS) LSTSQ070
000076          DO 30 K=1,NRDR                                          LSTSQ071
000077          DO 30 I=1,NRDR                                          LSTSQ072
000100          IF(I+K-2)27,28,27                                       LSTSQ073
000102       27 INSRT = I+K-2                                           LSTSQ074
000105          MATRX(K,I) = SX(INSRT)                                  LSTSQ075
000112          GO TO 30                                                LSTSQ076
000113       28 MATRX(K,I) = NUM                                        LSTSQ077
000117       30 CONTINUE                                                LSTSQ078
                C SOLVE NORMAL EQUATIONS(MATRIX)                        LSTSQ079
                C     GAUSS ELIMINATION SEQUENCE                        LSTSQ080
000124          LITE=2                                                  LSTSQ081
000125          DO110 J=1,NRDR                                          LSTSQ082
000126          JI=J+1                                                  LSTSQ083
000130          IF(JI-NRDR)6,6,5                                        LSTSQ084
000132       5  LITE=1                                                  LSTSQ085
000133       6  CONTINUE                                                LSTSQ086
000133          L=J                                                     LSTSQ087
000135          DO 12 I=JI,NRDR                                         LSTSQ088
000137          GO TO (12,17),LITE                                      LSTSQ089
000145       17 IF(ABS(MATRX(I,J))-ABS(MATRX(L,J)))12,12,111            LSTSQ090
000160      111 L=I                                                     LSTSQ091
000162       12 CONTINUE                                                LSTSQ092
000165          IF(L-J)13,14,13                                         LSTSQ093
000167       13 DO 16 K=J,NRDR                                          LSTSQ094
000171          SWIT=MATRX(J,K)                                         LSTSQ095
000175          MATRX(J,K)=MATRX(L,K)                                   LSTSQ096
000203       16 MATRX(L,K)=SWIT                                         LSTSQ097
000211          SWIT=YSUM(J)                                            LSTSQ098
000213          YSUM(J)=YSUM(L)                                         LSTSQ099
000216          YSUM(L)=SWIT                                            LSTSQ100
000220       14 CONTINUE                                                LSTSQ101
000220          DO110 I=JI,NRDR                                         LSTSQ102
000222          GO TO(110,23),LITE                                      LSTSQ103
000230       23 EM=MATRX(I,J)/MATRX(J,J)                                LSTSQ104
000237          MATRX(I,J)=0.                                           LSTSQ105
000241          DO 15 K=1,NRDR                                          LSTSQ106
000243          C=EM*MATRX(J,K)                                         LSTSQ107
000250          MATRX(I,K)=MATRX(I,K)-C                                 LSTSQ108
000255       15 CONTINUE                                                LSTSQ109
000257      110 YSUM(I)=YSUM(I)-EM*YSUM(J)                              LSTSQ110
000271          RETRN(NRDR)=YSUM(NRDR)/MATRX(NRDR,NRDR)                 LSTSQ111
000277          IN=NRDR+1                                               LSTSQ112
000301          DO 41 NIX=2,NRDR                                        LSTSQ113
000302          MIX=IN-NIX                                              LSTSQ114
000304          MIN=MIX+1                                               LSTSQ115
000306          SUMA=0.                                                 LSTSQ116
```

```
000311      130 SUMA=-MATRX(MIX,KIX)*RETRN(KIX)+SUMA                    LSTSQ118
000323          RETRN(MIX)=(YSUM(MIX)+SUMA)/MATRX(MIX,MIX)              LSTSQ119
000332       41 CONTINUE                                                LSTSQ120
000334       42 RETURN                                                  LSTSQ121
000335          END                                                     LSTSQ122
SUBPROGRAM LENGTH
001316

UNUSED COMPILER SPACE
026400
```

LOAD MAP FILE = LGO 10.23.20. 03/15/70. PAGE 1

FWA LOAD 100 LWA LOAD 27407 FWA LOADER 04277 FWA TABLES 63512 UNUSED STORAGE 34103

PROGRAM	ADDRESS	FILE	COMMON	ADDRESS	LENGTH
LEAST	100	LGO			
LSTSQ	20727	LGO			
ACGOER	22245	SYSTEM			
INPUTC	22257	SYSTEM			
SYSTEM	23400	SYSTEM	SCOPE2	23400	
INPUTS	24451	SYSTEM			
OUTPTC	24533	SYSTEM			
RBAIEX	26073	SYSTEM			
FATAL7R	26134	SYSTEM			
SIOS	26252	SYSTEM			
GETBA	27370	SYSTEM			
			/BLANK/	0	0

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
LEAST	101								
LSTSQ	20730	LEAST	663	745	1027	1111	1173	1255	1337	1421
			1503	1565						
ACGOER	22246	LSTSQ	142	225						
INPUTC	22261	LEAST	6	10	11	36	40	41	54	56
			57	62	64	65	102	104	105	110
			112	113	132	134	135	140	142	143
			625	627	630	707	711	712	771	773
			774	1053	1055	1056	1135	1137	1140	1217
			1221	1222	1301	1303	1304	1363	1365	1366
			1445	1447	1450	1527	1531	1532		
KRAKER	22363	INPUTS	5	23						
QBNTRY	23401	LEAST	2							
SYSTEM	23606	ACGOER	3							
		INPUTC	37	727						
		INPUTS	26							
		OUTPTC	14	1144						
		RBAIEX	23							
SYSTEMC	23552									
SYSTEMP	23601									
END	23475	LEAST	1612							
		LSTSQ	336							
STOP	23525	LEAST	1610							
EXIT	23517									
ABNORML	23535	ACGOER	4							
		INPUTC	40	730						
		INPUTS	27							
		OUTPTC	15	1145						

Continued on following page

INPUTS	24453	LEAST	14	16	20	22	24	26	30	32
			33	44	46	50	51	70	72	74
			76	77	116	120	122	124	126	127
			146	150	152	153	277	301	303	304
			307	311	313	314	324	326	330	331
			334	336	340	341	351	353	355	356
			361	363	365	366	376	400	402	403
			406	410	412	413	423	425	427	430

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
		433	435	437	440	450	452	454	455
		460	462	464	465	475	477	501	502
		505	507	511	512	522	524	526	527
		532	534	536	537	547	551	553	554
		557	561	563	564	574	576	600	601
		604	606	610	611				
OUTPTC	24535	LEAST 633	635	636	641	645	647	651	653
		657	666	670	671	674	700	704	715
		717	720	723	727	731	733	735	741
		750	752	753	756	762	766	777	1001
		1002	1005	1011	1013	1015	1017	1023	1032
		1034	1035	1040	1044	1050	1061	1063	1064
		1067	1073	1075	1077	1101	1105	1114	1116
		1117	1122	1126	1132	1143	1145	1146	1151
		1155	1157	1161	1163	1167	1176	1200	1201
		1204	1210	1214	1225	1227	1230	1233	1237
		1241	1243	1245	1251	1260	1262	1263	1266
		1272	1276	1307	1311	1312	1315	1321	1323
		1325	1327	1333	1342	1344	1345	1350	1354
		1360	1371	1373	1374	1377	1403	1405	1407
		1411	1415	1424	1426	1427	1432	1436	1442
		1453	1455	1456	1461	1465	1467	1471	1473
		1477	1506	1510	1511	1514	1520	1524	1535
		1537	1540	1543	1547	1551	1553	1555	1561
		1570	1572	1573	1576	1602	1606		
KODER	24674								
RBAIEX	26074	LSTSQ	43	57					
FATAL78	26135	SYSTEM	710						
CARD.F	24232	SIOS	56						
BKSPRU.	26541								
FIZBAK.	26551								
POSFIL.	26577	OUTPTC	32						
RDPRU.	26667								
DAT.	26710	OUTPTC	56	72	37				
CIOI.	26515								
OPEN.	26254	INPUTC	22						
		SYSTEM	506						
		OUTPTC	22						
SIO.	26373	INPUTC	52						
		OUTPTC	71						
ADVIN.	26610	SYSTEM	410						
MVWDS.	26422								
POSFI.	26620								
FIZBA.	26632								
DAT..	27140	INPUTC	56	24	47					
		INPUTS	17	33	42	45				
GETBA	27370	INPUTC	10							
		OUTPTC	10							

UNSATISFIED EXTERNALS REFERENCES (RELATIVE)

** NONE **

STRAIGHT LINE LEAST SQUARES FIT THROUGH 137 POINTS (X,Y)

A- 7(	18,	29)	A- 7(	15,	15)	A- 7(	13,	9)	A- 7(	6,	3)	A- 13(	19,	37)	A- 13(	16,	17)
A- 13(	16,	16)	A- 13(	17,	23)	A- 13(	17,	19)	A- 13(	17,	24)	A- 13(	18,	21)	A- 13(	16,	9)
A- 14(	14,	11)	A- 14(	18,	18)	A- 14(	18,	30)	A- 14(	18,	23)	A- 14(	16,	16)	A- 14(	14,	11)
A- 14(	16,	19)	A- 21(	18,	22)	A- 22(	8,	17)	A- 22(	17,	25)	A- 22(	11,	19)	A- 22(	12,	24)
A- 22(	15,	26)	A- 22(	14,	32)	A- 39(	16,	26)	A- 47(	7,	13)	A- 47(	20,	14)	A- 47(	8,	21)
A- 47(	19,	4)	A- 47(	19,	10)	A- 47(	10,	6)	A- 50(	11,	12)	A- 52(	14,	21)	A- 52(	16,	22)
A- 59(	16,	11)	A- 64(	17,	18)	A-218(	18,	14)	A-218(	17,	11)	A-218(	17,	13)	A-218(	17,	14)
A-218(	9,	3)	A-218(	14,	7)	A-218(	11,	6)	A-218(	14,	8)	A-218(	10,	4)	A-223(	12,	3)
A-223(	15,	10)	A-223(	16,	7)	A-223(	16,	8)	A-223(	14,	7)	A-223(	17,	9)	A-223(	8,	10)
A-223(	16,	15)	A-223(	16,	9)	A-223(	16,	10)	A-223(	11,	3)	A-223(	14,	4)	A-223(	14,	4)
A-223(	16,	7)	A-223(	17,	9)	A-223(	18,	8)	A-223(	12,	7)	A-223(	19,	10)	A-223(	13,	8)
A-223(	16,	9)	A-225(	11,	10)	A-225(	8,	10)	A-225(	10,	7)	A-225(	9,	9)	A-225(	15,	12)
A-225(	10,	13)	A-225(	13,	7)	A-225(	11,	12)	A-225(	12,	12)	A-236(	6,	2)	A-236(	6,	1)
A-236(	10,	5)	A-236(	12,	7)	A-236(	10,	6)	A-236(	10,	7)	A-236(	13,	8)	A-236(	15,	10)
A-236(	6,	4)	A-236(	7,	3)	A-236(	10,	6)	A-237(	14,	4)	A-237(	16,	14)	A-237(	15,	7)
A-238(	13,	13)	A-238(	14,	11)	A-238(	16,	18)	A-257(	18,	52)	A-400(	19,	40)	A-404(	9,	12)
A-404(	14,	22)	A-404(	11,	11)	A-404(	13,	17)	A-404(	15,	10)	A-404(	13,	27)	A-404(	15,	19)
A-404(	14,	18)	A-404(	15,	11)	A-404(	15,	12)	A-404(	13,	15)	A-404(	18,	16)	A-404(	16,	17)
A-407(	19,	20)	A-407(	16,	6)	A-407(	17,	14)	A-407(	17,	7)	A-407(	17,	18)	A-407(	19,	38)
A-410(	14,	8)	A-410(	14,	7)	A-410(	19,	7)	A-412(	16,	8)	A-413(	18,	39)	A-416(	17,	6)
A-417(	17,	7)	A-419(	20,	25)	A-419(	15,	12)	A-422(	17,	7)	A-423(	18,	8)	A-440(	17,	40)
A-440(	17,	40)	A-622(	18,	20)	A-622(	19,	11)	A-622(	17,	48)	A-622(	21,	24)	A-622(	14,	5)
A-622(	14,	5)	A-629(	9,	3)	A-629(	9,	4)	A-629(	16,	9)	A-629(	17,	10)			

(B15/X VERSUS A1/Y)

Y = -5.121 + 1.302 * X

MF14330. 03/15/70.PURDUE MACE 02/07/70.

```
09.40.10.MF143/     15073,PERI.OFF,T60,TP1,CM65000.
09.40.10.P10.
09.40.10.MAP(ON)
09.40.10.RUN(S)
09.40.16.CTIME 002.814 SEC.  RUN MOD LEVEL 48
09.40.17.REQUEST(TAPE1,556,HY,X,C=84,MT,READ)
10.23.15.  MT51 ASSIGNED - 556
10.23.15.REWIND(TAPE1)
10.23.18.LGO.
10.23.20.CX     3.363 SEC.
10.23.20.PX     5.341 SEC.
10.23.20.NL    27600
10.37.16.STOP
10.37.16.CP    62.454 SEC.
10.37.16.PP   339.266 SEC.
10.37.16.LINES =          1376 OCTAL
10.37.16.CM     1.321 MWD-SEC.
```

HISTOGRAM ROUTINE

for non-blank entries for both variables
in the same data set

```
                PROGRAM CH2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE8)
000003          DATA BL/1H /
000003          DIMENSION BB15(49),DD26(49)
000003          N = 0
000004          DO 100 K=1,2170
000006          READ(1,3) A
000013          READ(1,1) B15
000021          READ(1,3) C
000027          READ(1,2) D26
000035          READ(1,3) E
000043          READ(1,3) F
000051          READ(1,3) G
000057          READ(1,3) H
000065      1   FORMAT(28X,A3)
000065      2   FORMAT(61X,A4)
000065      3   FORMAT(50X,A5)
000065          IF((B15.NE.BL).AND.(D26.NE.BL)) GO TO 10
000075          GO TO 100
000075     10   N = N + 1
000077          BB15(N) = B15
000101          DD26(N) = D26
000102     100  CONTINUE
000104          WRITE(8,60) (BB15(K),K=1,N)
000117      60  FORMAT(2X,A3)
000117          WRITE(8,50) (DD26(K),K=1,N)
000132      50  FORMAT(2X,A4)
000132          END FILE 8
000134          END

PROGRAM LENGTH INCLUDING I/O BUFFERS
004465

UNUSED COMPILER SPACE
027200

        7
        8
        9
PROBLM 1JUDD001000490010000                                          0000801
(2X,F3.1)
SELECT11000010.001
        HISTOGRAM FOR PROPAGATION VELOCITY, LONGITUDINAL WAVE/LAB
PROBLM 2JUDD001000490010000                                          0000801
(2X,F4.2)
SELECT11000010.001
                HISTOGRAM FOR YOUNG/S MODULUS, LAB STATIC (1ST LOADING)
FINISH
        6
        7
        8
        9
```

N.B. This insert indicates the deck set-up as submitted to the computer. These data cards for the BMD routine are not included in the output listing.

```
LOAD MAP      FILE = LGO                                  18.22.55. 12/19/69.   PAGE  1

FWA LOAD      100   LWA LOAD    11465   FWA LOADER  62303   FWA TABLES  64001   UNUSED STORAGE  52114
```

Continued on following page

```
CM2             100   L60
ENDFIL         4505   SYSTEM
INPUTC         4660   SYSTEM
SYSTEM         5001   SYSTEM
OUTPTC         7052   SYSTEM
FATAL/R       10412   SYSTEM
SIOS          10530   SYSTEM
GETMA         11046   SYSTEM

                                    SCOPE2          6001

                                    /BLANK/            0          0


ENTRY       ADDRESS         REFERENCES (RELATIVE)

CM2             101
ENDFIL         4506         CM2             133
INPUTC         4662         CM2               7          11         12         15         17         20         23         26
                                                26          31         33         34         37         41         42         45
                                                47          50         53         55         56         61         63         66

SHAKER         4764
SHNTRY         6002         CM2               2
SYSTEM         6207         ENDFIL           61
                            INPUTC           37        7577
                            OUTPTC           14        1144

SYSTEMC        6153
SYSTEMP        6202
END            6076         CM2             135
STOP           6126
EXIT           6120
ABNORML        6136         ENDFIL           62
                            INPUTC           40        7536
                            OUTPTC           15        1146
OUTPTC         7154         CM2             106         112        116        121        125        131
KODER          7213
FATAL7R       10413         SYSTEM          710
CARD.F        10510         SIOS             56
SKSPRV.       11017
FIZBAS.       11027
POSFIL.       11055         ENDFIL           45
                            OUTPTC           32

RDBNJ.        11145
DAT.          11166         OUTPTC           56         77         37
CIOL.         10773         ENDFIL           37         52         15
OPEN.         10532         ENDFIL           11
                            INPUTC           22
                            SYSTEM          506
                            OUTPTC           22
SIO.          11051         INPUTC           52
                            OUTPTC           71
ADVIN.        11066         ENDFIL           30
                            SYSTEM          410
MVBUS.        10700
POSFI.        11076         ENDFIL           24

LOAD MAP        FILE = L60
ENTRY       ADDRESS         REFERENCES (RELATIVE)                                  18.22.55. 12/10/69.   PAGE  2

FIZBA.        11110
DAT..         11016         INPUTC           56         74         47
GETMA         11046         ENDFIL            5
                            INPUTC           10
                            OUTPTC           10

UNSATISFIED EXTERNALS       REFERENCES (RELATIVE)
** NONE **
```

LOAD MAP FILE = LGO 18.59.25. 12/10/69. PAGE 1

FWA LOAD 100 LWA LOAD 26125 FWA LOADER 64303 FWA TABLES 63255 UNUSED STORAGE 5130

PROGRAM	ADDRESS	FILE
BMD5D	100	LGO
HIST	5645	LGO
PLOTR	6404	LGO
TRANS	7572	LGO
VFCHCK	10526	LGO
SCALE	10564	LGO
FILL	11026	LGO
FORM3	11071	LGO
ACOSEC	11222	SYSTEM
ALOG10	11234	SYSTEM
ATAN	11323	SYSTEM
EXP	11417	SYSTEM
INPUTC	11476	SYSTEM
SYSTEM	12617	SYSTEM
OUTPTC	13670	SYSTEM
OUTPTS	15230	SYSTEM
RBAIEX	15314	SYSTEM
SQRT	15355	SYSTEM
IBAIEX	15420	SYSTEM
RBAREX	15451	SYSTEM
FATAL78	15530	SYSTEM
SIOS	15546	SYSTEM
GETBA	16764	SYSTEM

COMMON	ADDRESS	LENGTH
PRESET	6400	
SCOPE2	12617	
/BLANK/	17003	37142

ENTRY	ADDRESS
BMD5D	101
HIST	5646
PLOTR	6406
TRANS	7573
VFCHCK	10527
SCALE	10566
FILL	11027
FORM3	11072
ACOSEC	11223
ALOG	11240
ALOG10	11235
ATAN	11324
EXP	11420
INPUTC	11500
RBAREX	11602
OBNTRY	12620

REFERENCES (RELATIVE)								
BMD5D	772							
BMD5D	626	631	644	647				
BMD5D	217							
BMD5D	76							
PLOTR	56	61	335	340				
BMD5D	371							
PLOTR	432	514						
TRANS	152							
TRANS	236							
SCALE	37							
RBAREX	17							
TRANS	17							
TRANS	247							
RBAREX	22							
BMD5D	13	15	17	21	23	25	27	31
	33	35	39	161	165	151	166	172
	174	264	266	270	272	274	275	300
	301	313	317	323	344	346	352	356
	360							
TRANS	45	47	51	53	55	57	60	
BMD5D	2							

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
SYSTEM	13665	ACGOEN	3						
		ALNLOG	40	42					
		ATAN	37						
		EXP	33						
		INPUTC	37	727					
		OUTPTC	14	1144					
		OUTPTS	46						
		RBAIEX	23						
		SQRT	25						
		IBAIEX	17						
		RBAREX	37						
SYSTEMC	12771								
SYSTEMP	13020								
END	12714	BMUSD	777						
		HIST	342						
		PLOTR	713						
		TRANS	510						
		VFCHCK	17						
		SCALE	170						
		FILL	32						
		FORM3	57						
STOP	12744	BMUSD	47						
EXIT	12736								
ABNORML	12754	ACGOEN	4						
		INPUTC	40	730					
		OUTPTC	15	1145					
		OUTPTS	47						
OUTPTC	13672	BMUSD	44	46	101	102	105	107	111	113
			115	117	121	123	124	444	445	453
			455	461	464	467	470	473	477	503
			507	511	515	521	524	525	545	547
			553	557	504	565	570	574	600	641
			662	673	675	676	712	716	722	727
			730	732	737	743	762	764	765	
		HIST	25	30	32	33	75	101	107	112
			155	167	100	271	273	275	303	307
			311	313	322	326	334	337		
		PLOTR	101	103	123	127	135	140	205	211
			217	222	235	235	251	255	263	267
			275	301	305	537	541	543	554	560
			562	563	626	630	632	640	644	646
			647	662	664	672	676	677		
		TRANS	31	32	35	36	71	73	75	77
			101	103	104	125	126	454	460	461
			464	466	467	504	505			
		VFCHCK	6	7						
KOMER	14031	OUTPTS	5	22						
OUTPTS	15232	FILL	7	13	21	24				
RBAIEX	15315	SCALE	44							
SQRT	15356	TRANS	14	174	217	222	271	317	332	343
			345							
IBAIEX	15421	SCALE	27							
RBAREX	15452	TRANS	413							
FATAL/R	15531	SYSTEM	710							
CARD.F	15626	SIOS	54							
MASPRO.	15135									

ENTRY	ADDRESS	REFERENCES (RELATIVE)			
FIZHAA.	15145				
PUSFIL.	15173	OUTPTC	32		
RUDMJ.	16263				
DAT.	15304	OUTPTC	56	72	37
		OUTPTS	44	17	27
CIOI.	16111				
OPEN.	15550	INPUTC	22		
		SYSTEM	506		
		OUTPTC	22		
SIO.	15767	INPUTC	52		
		OUTPTC	71		
AJVIN.	16204	SYSTEM	410		
MVAUS.	15016				
PISFI.	15214				
FIZHA.	16226				
DAT..	16534	INPUTC	56	96	41
GET6A	16764	INPUTC	10		
		OUTPTC	10		

UNSATISFIED EXTERNALS	REFERENCES (RELATIVE)

** NONE **

BMD05D GENERAL PLOT - INCLUDING HISTOGRAM - VERSION 06 AUG. 19, 1964
HEALTH SCIENCES COMPUTING FACILITY,UCLA

PROBLEM CODE . . . 13000
NO. OF VARIABLES . . . 1
NO. OF CASES 49
NO. OF SELECTION CARDS 1
NO. OF VARIABLES ADDED 0
NO. OF TRNGEN CARDS . 0
NO. OF FORMAT CARDS . 1

BMD05D GENERAL PLOT - INCLUDING HISTOGRAM - VERSION 06 AUG. 19, 1964
HEALTH SCIENCES COMPUTING FACILITY,UCLA

HISTOGRAM OF VARIABLE 1

11.9000	9.1000	9.6000	12.1000	12.6000	9.7000	10.7000	8.9000	10.9000	11.1000
11.5000	13.3000	9.7000	12.5000	11.9000	11.8000	12.1000	13.3000	12.8000	11.2000
11.9000	12.0000	11.4000	11.4000	16.1000	13.0000	12.2000	11.8000	14.9000	14.0000
14.2000	14.5000	14.6000	14.4000	10.1000	11.1000	10.7000	15.1000	12.2000	12.0000
12.4000	12.4000	9.1000	10.2000	15.2000	9.9000	15.1000	7.4000	10.7000	

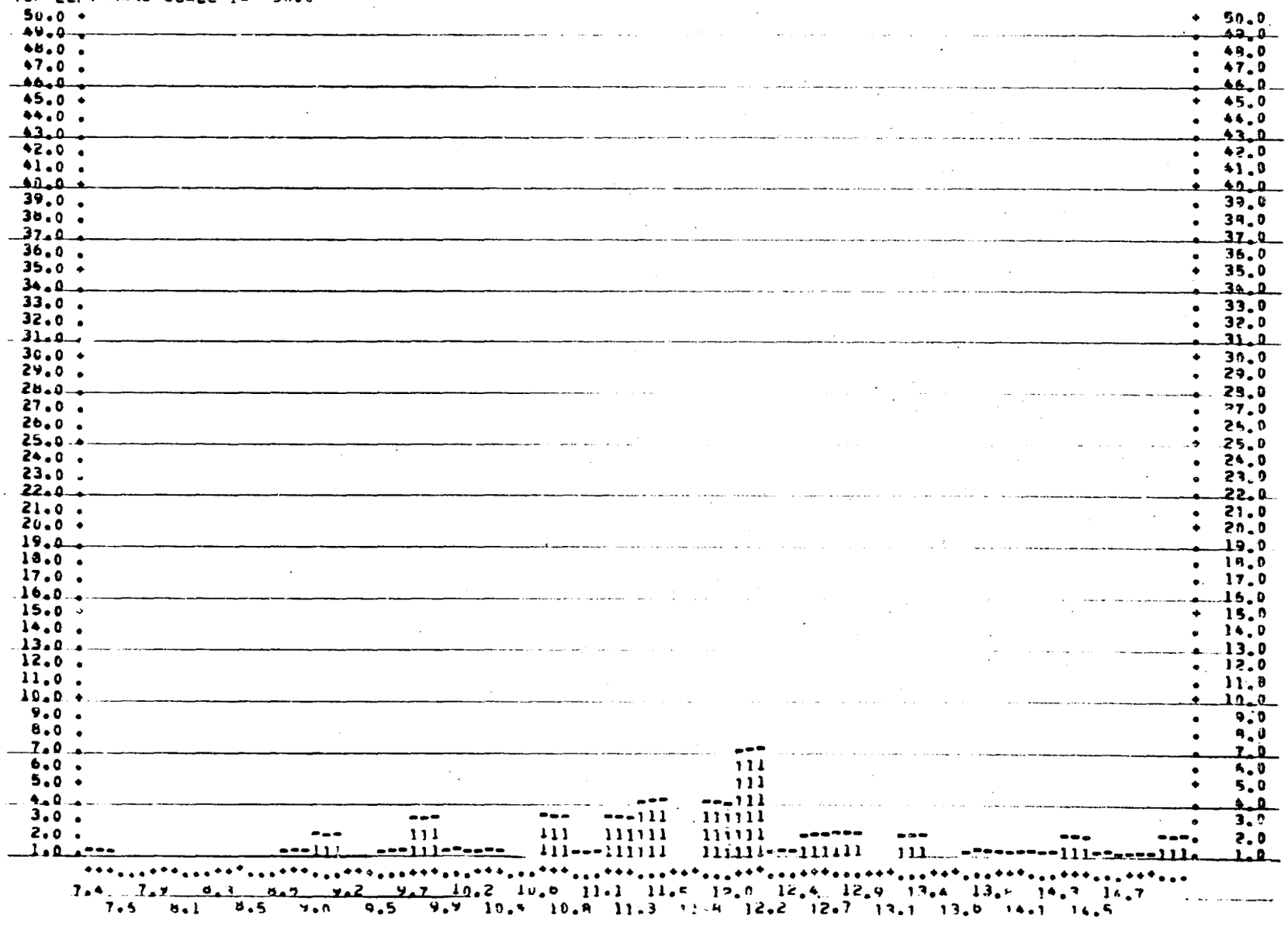
HISTOGRAM FOR PROPAGATION VELOCITY, LONGITUDINAL WAVE/LAB

THE VALUE GIVEN FOR THE INTERVAL WIDTH IS TOO SMALL.
A NEW VALUE, .2294, HAS BEEN SUBSTITUTED.

MIN = 7.379999 MAX = 15.200000

7.4 7.9 8.3 8.8 9.2 9.7 10.2 10.6 11.1 11.5 12.0 12.4 12.9 13.4 13.8 14.3 14.7
7.6 8.1 8.5 9.0 9.5 9.9 10.4 10.8 11.3 11.8 12.2 12.7 13.1 13.6 14.1 14.5

TOP LEFT HAND SCALE IS 50.0



7.4 7.9 8.3 8.8 9.2 9.7 10.2 10.6 11.1 11.5 12.0 12.4 12.9 13.4 13.8 14.3 14.7
7.6 8.1 8.5 9.0 9.5 9.9 10.4 10.8 11.3 11.8 12.2 12.7 13.1 13.6 14.1 14.5

BMD05D GENERAL PLOT - INCLUDING HISTOGRAM - VERSION 06 AUG. 19, 1964
HEALTH SCIENCES COMPUTING FACILITY,UCLA

PROBLEM CODE . . . 21000
NO. OF VARIABLES . . . 1
NO. OF CASES 49
NO. OF SELECTION CARDS 1
NO. OF VARIABLES ADDED 0
NO. OF INPUT CARDS . 0
NO. OF FORMAT CARDS . 1

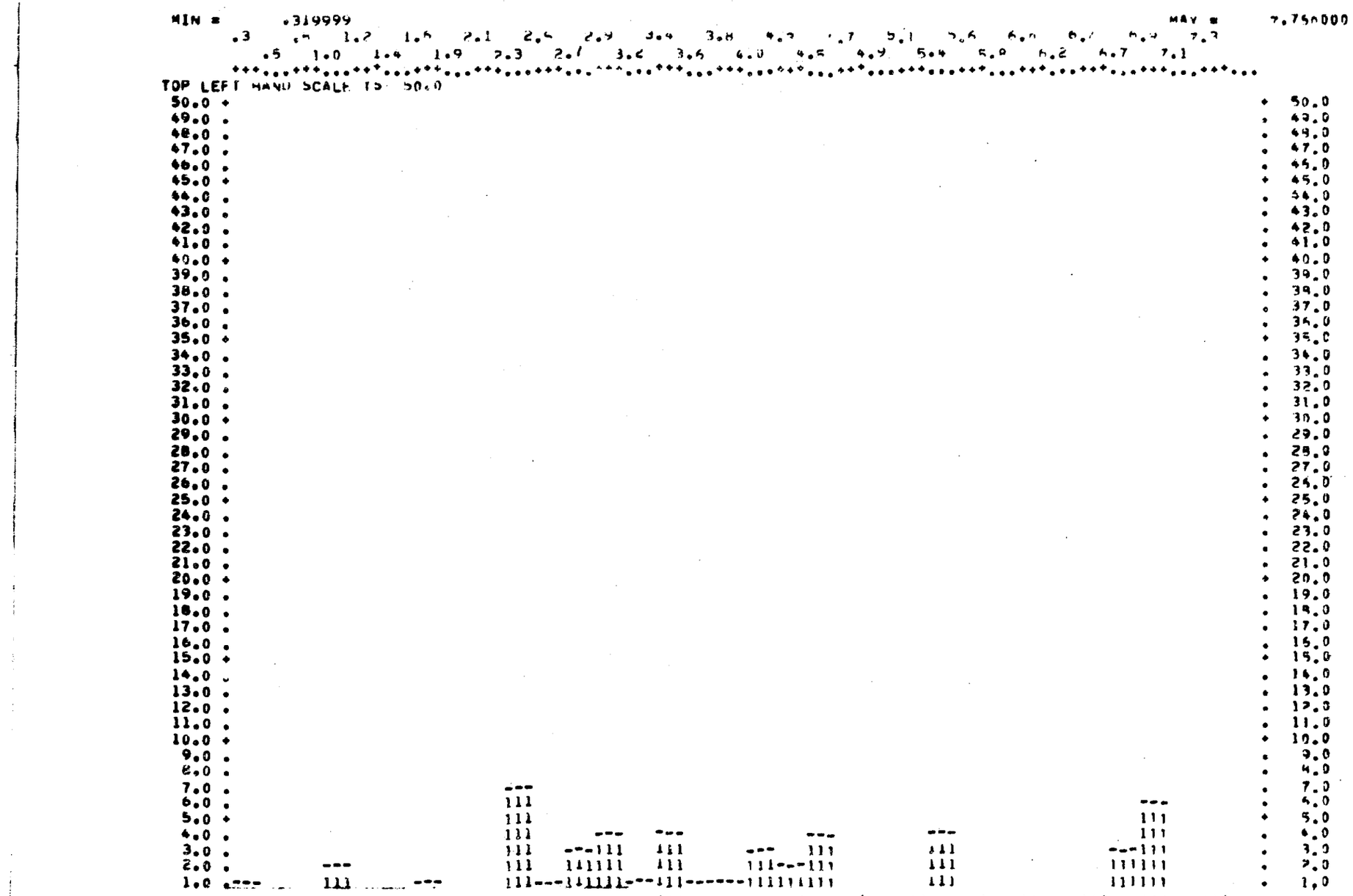
BMD05D GENERAL PLOT - INCLUDING HISTOGRAM - VERSION 06 AUG. 19, 1964
HEALTH SCIENCES COMPUTING FACILITY,UCLA

HISTOGRAM OF VARIABLE 1

3.4500	1.6800	2.5000	4.5100	4.3600	3.4900	2.7400	2.5600	3.2700	4.0000
3.5000	3.5500	2.5000	2.7400	2.4000	3.5000	4.1300	4.1000	4.1000	4.5400
4.5400	4.5400	2.4600	2.4900	2.4600	3.0300	3.0300	3.0300	6.9600	6.9400
6.9600	6.8000	6.8000	6.8000	2.5000	6.9400	6.9400	6.9400	5.3800	5.3800
5.3800	5.4800	2.9700	2.8600	1.1400	4.3100	7.7400	1.1400	.3200	

THE VALUE GIVEN FOR THE INTERVAL WIDTH IS TOO SMALL.
A NEW VALUE, .219, HAS BEEN SUBSTITUTED.

MIN = .319999 MAX = 7.750000



```
MM5394.   12/19/84.BURDUE PAGE 11/27/54.


09.29.55.MM5397     3512.NAMAS,T60.CM65000.TP1.P10
09.29.55..
09.29.55.MAP(ON)
09.29.55.RUN(S)
09.29.57.CTIME 000.208 SEC.  RUN MOD LEVEL 48
09.29.58.REQUEST(TAPE1,556,HY,X,C=R4,MT,REAL)
18.22.52.  MT50 ASSIGNED = 556
18.22.52.REWIND(TAPE1)
18.22.53.LGO.
18.22.55.CP      .460 SEC.
18.22.55.PP     3.496 SEC.
18.22.55.ML    12000
18.36.15.END     CM2
18.36.15.REWIND(TAPE1)
18.36.15.RETURN(TAPE1)
18.36.15.  TAPE LIMIT=0
18.36.15.REWIND(LGO)
18.36.15.REL(65000)
18.36.15.CP    20.124 SEC.
18.36.15.PP   341.041 SEC.
18.36.15.ML    65000
18.56.37.REWIND(TAPE4)
18.56.37.LINCOPY(STATSIN,LGO,RM050)
18.59.22.LGO.
18.59.27.CP    20.980 SEC.
18.59.27.PP   346.319 SEC.
18.59.27.ML    56300
18.59.30.STOP
18.59.30.CP    21.907 SEC.
18.59.30.PP   347.309 SEC.
18.59.30.LINES =          0430 OCTAL
18.59.30.CM     2.449 120-SEC.
```

BMD PROGRAM TO PRODUCE HISTOGRAM

```
SUBPROGRAM LENGTH
000052

TPWD2

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10     = 000007    12     = 000010    14     = 000012    15     = 000016
16     = 000014    18     = 000022    19     = 000020    22     = 000024
24     = 000026    28     = 000027    40     = 000030    49     = 000043

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

START OF CONSTANTS
000040

START OF TEMPORARIES
000050

START OF INDIRECTS
000052

UNUSED COMPILER SPACE
022400
```

```
      PROGRAM BMD05D (INPUT=1,OUTPUT=1,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8)      BMD5D00001
C     BMD05D      GENERAL PLOT WITH HISTOGRAM          AUGUST 18, 1964      BMD5D00002
C                       HEALTH SCIENCES COMPUTING FACILITY, UCLA             BMD5D00003
C                            6400 CONVERSION BETTY BENSON                    BMD5D00004
000003      DIMENSION FG(999),X(15000),SYM(15),NX(15),HEAD(17),YMT(500),XMA(50 BMD5D00005
           10),Z(15),NXA(15),AY(51,17)                                        BMD5D00006
C                                                                            BMD5D00007
000003      COMMON X,HEAD,XY,SYM,Z                                           BMD5D00008
000003      EQUIVALENCE (FG,AMT)                                             BMD5D00009
000003      TYPE INTEGER A123,B123,C123,D123,IODE,XY                         BMD5D00010
C                                                                            BMD5D00011
000003    110 FORMAT(70H1BMD05D GENERAL PLOT - INCLUDING HISTOGRAM - VERSION 06 BMD5D00012
          1 AUG. 18, 1964 /                                                  BMD5D00013
          140H HEALTH SCIENCES COMPUTING FACILITY,UCLA//)                    BMD5D00014
C                                                                            BMD5D00015
000003      A123=(*6HFINISH)                                                 BMD5D00016
000005      B123=(*6HPROBLM)                                                 BMD5D00017
000006      C123=(*6HCRSVAR)                                                 BMD5D00018
000010      D123=(*6HSELECT)                                                 BMD5D00019
000011    5 READ(            5,101)IODE,SAME,NV,NP,NG,NADD,NTRAN,MTAPE,NCARD BMD5D00020
000037    204 FORMAT(45H0CONTROL CARDS INCORRECTLY ORDERED OR PUNCHED)         BMD5D00021
000037      IF (IODE-A123) 200,201,200                                       BMD5D00022
000042    202 WRITE(             6,204)                                      BMD5D00023
000046    201 STOP                                                         BMD5D00024
000050    200 IF (IODE-B123) 202,203,202                                   BMD5D00025
000052    203 NTAPE=MTAPE                                                  BMD5D00026
000054      IF (NTAPE.EQ.0) NTAPE=5                                          BMD5D00027
000056    306 IF(NV*(NV-501))309,202,202                                   BMD5D00028
000062    309 IF((NP-1)*(NP-15001))205,202,202                             BMD5D00029
000070    205 IF((NV+NADD)*NP-15000) 206,206,202                           BMD5D00030
000075    206 CALL VECHCK(NCARD)                                           BMD5D00031
```

```
000077      207 WRITE(                6,110)                                        BMD5D00032
000103          WRITE(                6,210)SAME,NV,NP,NG,NADD,NTRAN,NCARD          BMD5D00033
000125       71 NTOT=NP*NV-NP                                                       BMD5D00034
000130          IF((NV+NADD)*(NV+NADD-50))1,202,202                                 BMD5D00035
000136        1 NCARD=NCARD*8                                                       BMD5D00036
000140          READ(                5,102)(FG(I),I=1,NCARD)                        BMD5D00037
000152          NCARD=NTOT+NP                                                      BMD5D00038
000154          DO 211 I=1,NCARD                                                   BMD5D00039
000156      211 X(I)=0.0                                                           BMD5D00040
000162       70 DO 3 I=1,NP                                                        BMD5D00041
            C       *****   READ IN THE RAW DATA AND TRANSPOSE THE MATRIX         BMD5D00042
000166          READ(                NTAPE,FG)(AMA(J), J=1,NV)                     BMD5D00043
000177          DO 3 J=1,NV                                                        BMD5D00044
000201          K=NP*J-NP+I                                                        BMD5D00045
000204        3 X(K)=XMA(J)                                                        BMD5D00046
000212          IF(NTRAN) 202,22,21                                                BMD5D00047
000214       21 CALL TRANS(NP,NV,NTRAN)                                            BMD5D00048
000217          IF(NV) 202,202,22                                                  BMD5D00049
000221       22 NPV=NP                                                             BMD5D00050
000223          IF (NADD) 999,999,998                                              BMD5D00051
000224      998 NV = NV + NADD                                                     BMD5D00052
000226      999 K = 1                                                              BMD5D00053
000227          DO 63 I=1,NV                                                       BMD5D00054
000231          XMI(I)=9999.999.                                                   BMD5D00055
000233          XMA(I)=-99999999.                                                  BMD5D00056
000235          DO 64 J=K,NPV                                                      BMD5D00057
000237          XMI(I)= AMIN1(X(J),XMI(I))                                         BMD5D00058
000244       64 XMA(I)= AMAX1(X(J),XMA(I))                                         BMD5D00059
000253          K=K+NP                                                             BMD5D00060
000255       63 NPV=NPV+NP                                                         BMD5D00061
000260          DO 50 JJ=1,NG                                                      BMD5D00062
000261          READ(                5,104)TODE,NH,NL,NC,NY,FN                     BMD5D00063
000300          IF (TODE-0123) 202,209,202                                         BMD5D00064
000302      209 IF(NH*(NH-4))215,202,202                                           BMD5D00065
000306      215 NH=NH*8                                                            BMD5D00066
000310          READ(                5,102)(HEAD(I),I=1,NH)                        BMD5D00067
000322          IF(NC) 202,20,8                                                    BMD5D00068
000324        8 NNC=(NC+6)/7                                                       BMD5D00069
000330          IF(NNC-2)9,9,202                                                   BMD5D00070
000332        9 NG2=0                                                              BMD5D00071
000333          DO 150 I=1,NNC                                                     BMD5D00072
000335          NG1=NG2+1                                                          BMD5D00073
000337          NG2=NG2+7                                                          BMD5D00074
000340          READ(                5,105)TODE,(NA(J),SYM(J),J=NG1,NG2)           BMD5D00075
000357          IF (TODE-0123) 202,150,202                                         BMD5D00076
000361      150 CONTINUE                                                           BMD5D00077
            C                                                                      BMD5D00078
            C       BATTELLE COMMENT                                               BMD5D00079
            C       FILL COMPUTER WORD WITH INPUT SYMBOL                           BMD5D00080
            C                                                                      BMD5D00081
000364          IF(NC.GE.1) CALL FILL(SYM, NC)                                     BMD5D00082
            C                                                                      BMD5D00083
000370          XMAA=-99999999.                                                    BMD5D00084
000372          XMIN=99999999.                                                     BMD5D00085
000373          IF(NC-1)20,11,12                                                   BMD5D00086
000376       11 J=NA(1)                                                            BMD5D00087
000400          XMAX=XMA(J)                                                        BMD5D00088
000402          XMIN=XMI(J)                                                        BMD5D00089
000403          GO TO 14                                                           BMD5D00090
000404       12 DO 13 I=1,NC                                                       BMD5D00091
000406          J=NA(I)                                                            BMD5D00092
000410          XMAX= AMAX1(XMAX,XMA(J))                                           BMD5D00093
```

```
000414      13 XMIN= AMIN1(XMIN,AMT(J))                                          BMD5D00095
000423      14 NPV=0                                                             BMD5D00095
000424      10 DO 65 I=1,NC                                                      BMD5D00096
000426      65 NXX(I)=NX(I)*NP-NP                                                BMD5D00097
000434         NYY=NY*NP-NP                                                      BMD5D00098
000436         IF(NL)23,23,24                                                    BMD5D00099
000440      24 WRITE(           6,110)                                           BMD5D00100
000444         IF(9-NC)242,249,249                                               BMD5D00101
000447     242 WRITE(           6,108)NY,(NX(I),I=1,9)                           BMD5D00102
000463         WRITE(           6,112)                                           BMD5D00103
000467         WRITE(           6, 111)(NX(I),I=10,NC)                           BMD5D00104
000502         GO TO 250                                                         BMD5D00105
000503     249 WRITE(           6,108)NY,(NX(I),I=1,NC)                          BMD5D00106
000520     250 WRITE(           6,112)                                           BMD5D00107
000524         DO 26 I=1,NP                                                      BMD5D00108
000526         MY=NYY+I                                                          BMD5D00109
000530         Y=X(MY)                                                           BMD5D00110
000532         DO 25 J=1,NC                                                      BMD5D00111
000533         MX=NXX(J)+I                                                       BMD5D00112
000535      25 Z(J)=X(MX)                                                        BMD5D00113
000541      26 WRITE(           6,106)Y,(Z(K),K=1,NC)                            BMD5D00114
         C  23 WRITE OUTPUT TAPE 6,110                        (ORIGINAL)       BMD5D00115
000561         WRITE (6,7777)                                                    BMD5D00116
000564      23 WRITE(           6,103)(HEAD(I),I=1,NH)                           BMD5D00117
000577         NNP=FN                                                            BMD5D00118
000601         YMAX=XMA(NY)                                                      BMD5D00119
000603         YMIN=XMI(NY)                                                      BMD5D00120
000605         DO 16 I=1,NP                                                      BMD5D00121
000606         MY=NYY+I                                                          BMD5D00122
000610         Y=X(MY)                                                           BMD5D00123
000612         DO 15 J=1,NC                                                      BMD5D00124
000613         MX=NXX(J)+I                                                       BMD5D00125
000615      15 Z(J)=X(MX)                                                        BMD5D00126
000621      16 CALL PLOTR(Y,YMIN,YMAX,XMIN,XMAX,NC,NNP)                          BMD5D00127
000633         IF(NNP)31,32,32                                                   BMD5D00128
000634      31 NC=-1                                                             BMD5D00129
000635         GO TO 33                                                          BMD5D00130
000636      32 NC=0                                                              BMD5D00131
000637      33 CALL PLOTR(Y,YMIN,YMAX,XMIN,XMAX,NC,NNP)                          BMD5D00132
000646         GO TO 50                                                          BMD5D00133
000647      20 NYT=NY*NP                                                         BMD5D00134
000652         NYY=NYT-NP+1                                                      BMD5D00135
000654         IF(NL)29,29,28                                                    BMD5D00136
000655      28 WRITE(           6,110)                                           BMD5D00137
000661         NNC=(NP+9)/10                                                     BMD5D00138
000665         NG2=NYY-1                                                         BMD5D00139
000667         WRITE(           6, 107)NY                                        BMD5D00140
000675         DO 285 I=1,NNC                                                    BMD5D00141
000677         NG1=NG2+1                                                         BMD5D00142
000701         NG2=NG2+10                                                        BMD5D00143
000702         IF(NYT-NG2)283,284,284                                            BMD5D00144
000704     283 NG2=NYT                                                          BMD5D00145
000706     284 WRITE(           6,105)(X(J),J=NG1,NG2)                           BMD5D00146
000721     285 CONTINUE                                                         BMD5D00147
         C  29 WRITE OUTPUT TAPE 6,110                        (ORIGINAL)       BMD5D00148
000724         WRITE (6,7777)                                                    BMD5D00149
000727      29 WRITE(           6,103)(HEAD(I),I=1,NH)                           BMD5D00150
000742         XMAX=XMA(NY)                                                      BMD5D00151
000744         XMIN=XMI(NY)-.0000005                                             BMD5D00152
000746         IF((XMAX-XMIN)/FN-34.)34,34,35                                    BMD5D00153
000753      35 FN=(XMAX-XMIN)/34.                                                BMD5D00154
000756         WRITE(           6,109)FN                                         BMD5D00155
000764         GO TO 34                                                          BMD5D00156
```

```
000765      36 CALL HIST(NYY,NYT,YMIN,XMAA,FN,MP)                          BMD5D00157
000771      50 CONTINUE                                                    BMD5D00158
000774         GO TO 5                                                     BMD5D00159
           C                                                              BMD5D00160
000774     101 FORMAT(2A6,I3,I5,I3,I4,3BX,I3,212)                         BMD5D00161
000774     102 FORMAT (8A10)                                              BMD5D00162
000774     103 FORMAT (1H0,20X,8A10)                                      BMD5D00163
000774    7777 FORMAT (1H1)                                               BMD5D00164
000774     104 FORMAT(A6,2I1,I2,I3,F11.0)                                 BMD5D00165
           C      HATTELLE COMMENT                                        BMD5D00166
           C      FORMAT 105 CHANGED FROM (7(A6,I3),A6)                   BMD5D00167
           C      IN ORDER TO COMPLIMENT ADDED BATTELLE SUBROUTINE CALLED FILL.  BMD5D00168
000774     105  FORMAT(A6, /(I3, A1, 5X))                                 BMD5D00169
000774     106 FORMAT(1H 10(F10.4,1X))                                    BMD5D00170
000774     107 FORMAT(1H 23H HISTOGRAM OF VARIABLE I3//)                  BMD5D00171
000774     108  FORMAT(14H BASE VARIABLE, 3X,16H CROSS VARIABLES/6X,10(I3,8X))  BMD5D00172
000774     109 FORMAT(1H ,54H THE VALUE GIVEN FOR THE INTERVAL WIDTH IS TOO SMALL  BMD5D00173
              1, /13H A NEW VALUE,F11.4,22H,HAS BEEN SUBSTITUTED.//)       BMD5D00174
000774     111 FORMAT(5X,5(I3,8X))                                        BMD5D00175
000774     112 FORMAT(1H )                                                BMD5D00176
000774     125  FORMAT(1H 10F11.4)                                        BMD5D00177
000774     210  FORMAT(14H PROBLEM CODE 3(2H. )1AA6,/18H NO. OF VARIABLES 3(2H. ),  BMD5D00178
              A13,/14H NO. OF CASES 4(2H. )I5,/24H NO. OF SELECTION CARDS I3,/24H  BMD5D00179
              A NO. OF VARIABLES ADDED I3,/22H NO. OF TRNSFN CARDS  2H. I3,/22H N  BMD5D00180
              AO. OF FORMAT CARDS  2H. I3,//)                              BMD5D00181
           C                                                              BMD5D00182
000774         END                                                        BMD5D00183

PROGRAM LENGTH INCLUDING I/O BUFFERS
005544
```

```
BMD5D

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1     - 000136   5     - 000011   8     - 000324   9     - 000332
10    - 000424   11    - 000376   12    - 000404   14    - 000423
20    - 000647   21    - 000214   22    - 000221   23    - 000564
24    - 000440   28    - 000655   29    - 000727   31    - 000634
32    - 000636   33    - 000637   34    - 000765   35    - 000753
50    - 000771   65    - 000427   70    - 000162   71    - 000125
101   - 001061   102   - 001065   103   - 001067   104   - 001074
105   - 001100   106   - 001104   107   - 001107   108   - 001114
109   - 001123   110   - 001007   111   - 001140   112   - 001143
125   - 001145   150   - 000341   200   - 000050   201   - 000046
202   - 000042   203   - 000052   204   - 001043   205   - 000070
206   - 000075   207   - 000077   209   - 000302   210   - 001150
211   - 000157   215   - 000306   242   - 000447   249   - 000503
250   - 000520   283   - 000704   284   - 000706   304   - 000056
309   - 000062   998   - 000224   999   - 000226   7777  - 001072

BLOCK NAMES AND LENGTHS
      - 037122
```

```
VARIABLE ASSIGNMENTS
A123   =  004207    B123   =  004210    C123   =  004211    D123   =  004212
FB     =  001216    FN     =  004237    HEAD   =  035230C01 I      =  004226
J      =  004227    JJ     =  004232    K      =  004230    MTAPE  =  004222
MX     =  004250    MY     =  004246    NADD   =  004220    NC     =  004235
NCARD  =  004223    NG     =  004217    NG1    =  004246    NG2    =  004241
NH     =  004233    NL     =  004234    NNC    =  004240    NNP    =  004251
NP     =  004216    NPV    =  004231    NTAPE  =  004224    NTC1   =  004225
NTRAN  =  004221    NV     =  004215    NX     =  003165    NXX    =  004170
NY     =  004236    NY1    =  004254    NYY    =  004245    SAME   =  004214
SYM    =  037064C01 TODE   =  004213    X      =  000000C01 XMA    =  003204
XMAX   =  004243    X4I    =  001216    XMIN   =  004244    XY     =  035321C01
Y      =  004247    YMAX   =  004252    YMIN   =  004253    Z      =  037103C01

START OF CONSTANTS
000776

START OF TEMPORARIES
001200

START OF INDIRECTS
001212
UNUSED COMPILER SPACE
017200


      SUBROUTINE HIST(NYY,NYT,XMIN,XMAX,SYMB,NP)                          BMD50D0184
C     CHIST          SUBROUTINE HIST FOR BMD05D              JULY 17, 1964 BMD50D0185
000011      DIMENSION XY(51,17),X(15000),INT(35),XM(3),D(3),SYM(15),      BMD50D0186
     1     XZ(15),BONE(3),CLAB(57)                                        BMD50D0187
C                                                                         BMD50D0188
000011      COMMON X,CLAB,XY,SYM,Z                                        BMD50D0189
000011      TYPE INTEGER XM,BONE,D,XY,#                                   BMD50D0190
000011   23 FORMAT(1H F5.1,1X,A1,17A6,A1,1X,F5.1)                         BMD50D0191
000011  101 FORMAT(1H 5X,16(F4.1,2X),F4.1/9X,16(F4.1,2X)/8X,17(4H++++...)) BMD50D0192
000011  102 FORMAT(8X,17(6H+++...)/6X,16(F4.1,2X),F4.1/9X,16(F4.1,2X))    BMD50D0193
000011 4000 FORMAT( 8H  MIN = ,F12.6,8UX,7H MAX = ,F12.6)                 BMD50D0194
C     BATTELLE COMMENT                                                    BMD50D0195
C     OCTAL MASKS INCREASED FOR 6400 WORD SIZE.                           BMD50D0196
000011      XM(1) = 0000007777770000000008                                BMD50D0197
000012      XM(2) = 7777770000000000000008                                BMD50D0198
000014      BONE(1)=(+6H111000).AND.XM(2)                                 BMD50D0199
000016      BONE(2)=(+6H000111).AND.XM(1)                                 BMD50D0200
000020      D(1)=(+6H---000).AND.XM(2)                                    BMD50D0201
000022      D(2)=(+6H000---).AND.XM(1)                                    BMD50D0202
000024      M=1                                                           BMD50D0203
000025      WRITE(          6,4000)XMIN,XMAX                              BMD50D0204
000034      DO 50 I=1,35                                                  BMD50D0205
000041   50 INT(I)=0                                                      BMD50D0206
000044      DO 100 K=1,17                                                 BMD50D0207
000046      DO 100 J=1,50                                                 BMD50D0208
C     BATTELLE COMMENT                                                    BMD50D0209
C     HOLLERITH BLANKS USED INSTEAD OF 6060 OCTAL BLANKS.                 BMD50D0210
000047  100 XY(J,K) = (+6H      )                                         BMD50D0211
000057      MINH=XMIN/SYMB                                                BMD50D0212
000062      TXMIN=XMIN/SYMB-1.0                                           BMD50D0213
000064      CLAB(1)=XMIN                                                  BMD50D0214
000065      DO 16 I=2,34                                                  BMD50D0215
000066   16 CLAB(I)=CLAB(I-1)+SYMB                                        BMD50D0216
000073      WRITE(          6,101)(CLAB(I),I=1,34,2),(CLAB(J), J=2,33,2)  BMD50D0217
```

```
000112          DO 1 I=NYY,NYT                                                  BMD5D00218
000117          K=X(I)/SYMB-TXMIN                                               BMD5D00219
000123          INT(K)=INT(K)+1                                                 BMD5D00220
000127          IF(INT(M)-INT(K))8,1,1                                          BMD5D00221
000133        8 M=K                                                             BMD5D00222
000135        1 CONTINUE                                                        BMD5D00223
000140          YMAX=INT(M)                                                     BMD5D00224
000142          SC=50.0                                                         BMD5D00225
000144       32 IF(YMAX-SC)30,30,31                                             BMD5D00226
000147       31 SC=SC+50.0                                                      BMD5D00227
000151          GO TO 32                                                        BMD5D00228
              C                                                                 BMD5D00229
000152       30 WRITE(           6,103)SC                                       BMD5D00230
000160      103 FORMAT(23H TOP LEFT HAND SCALE IS F6.1)                         BMD5D00231
000160          SC=50.0/SC                                                      BMD5D00232
000162       15 DO 6 I=1,34                                                     BMD5D00233
000167          XL=INT(I)                                                       BMD5D00234
000171          L=XL*SC+.5                                                      BMD5D00235
000175          IF(L)5,6,5                                                      BMD5D00236
000176        5 MB=(3*I-1)/6+1                                                  BMD5D00237
000203          IT= MOD (I,2)                                                   BMD5D00238
000207          IF(IT)62,59,62                                                  BMD5D00239
000210       59 IT=2                                                            BMD5D00240
000211       62 XY(L,MB)=(XY(L,MB).AND.XM(IT)).OR.U(IT)                         BMD5D00241
000221          L=L-1                                                           BMD5D00242
000222          IF(L)11,6,11                                                    BMD5D00243
000223       11 DO 10 K=1,L                                                     BMD5D00244
000225       10 XY(K,MB)=(XY(K,MB).AND.XM(IT)).OR.BONE(IT)                      BMD5D00245
000244        6 CONTINUE                                                        BMD5D00246
000246          DO 7 K=1,50                                                     BMD5D00247
000250          L=51-K                                                          BMD5D00248
000252          R=L                                                             BMD5D00249
000253          R=R/SC                                                          BMD5D00250
000255          I= MOD (K,5)                                                    BMD5D00251
000261          IF(I-1)2,3,2                                                    BMD5D00252
000263        3 W=(+1H+)                                                        BMD5D00253
000265          GO TO 7                                                         BMD5D00254
000265        2 W=(+1H.)                                                        BMD5D00255
000267        7 WRITE(            6,23)R,W,(XY(L,M),M=1,17),W,R                 BMD5D00256
000321          WRITE(            6,102)(CLAH(I),I=1,34,2),(CLAB(J),I=2,33,2)   BMD5D00257
000340          RETURN                                                          BMD5D00258
000341          END                                                             BMD5D00259

SUBPROGRAM LENGTH
000536

MIST

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1       =  000135     2       =  000265     3       =  000263     5       =  000176
6       =  000244     7       =  000267     8       =  000133     10      =  000225
11      =  000223     15      =  000162     10      =  000066     23      =  000344
30      =  000152     31      =  000147     32      =  000144     50      =  000041
59      =  000210     62      =  000211     100     =  000047     101     =  000351
102     =  000360     103     =  000414     4000    =  000367

BLOCK NAMES AND LENGTHS
        =  037122
```

```
VARIABLE ASSIGNMENTS
BONE   = 000515    CLAB   = 035230C01 D      = 000512    I      = 000522
INT    = 000444    IT     = 000534    J      = 000524    K      = 000523
L      = 000532    M      = 000521    MB     = 000533    NUM    = 000525
R      = 000535    SC     = 000530    SYM    = 037064C01 TXMIN  = 000526
W      = 000520    X      = 000000C01 XL     = 000531    XM     = 000507
XY     = 035321C01 YMAX   = 000527    Z      = 037103C01

START OF CONSTANTS
000343

START OF TEMPORARIES
000426

START OF INDIRECTS
000440

UNUSED COMPILER SPACE
021200
```

```
                SUBROUTINE PLOTR(X,ZMIN,ZMAX,WMIN,WMAX,NC,VP)                    BMD50D0260
      CPLOTR          SUBROUTINE PLOTR FOR BMD09D (MODIFIED) JULY 20, 1964       BMD50D0261
      C       BATTELLE COMMENT                                                   BMD50D0262
      C       MASKING VECTOR (XM) REMOVED FROM PLOTR, NOW IN FORM3.              BMD50D0263
000012        DIMENSION V(25000),Y(15),XY(51,17),CLAB(57),XM(4),SYM(15)          BMD50D0264
             1  ,GF(10),FMT(10)                                                  BMD50D0265
000012        INTEGER BLANKS,GF,FMT,TC,TP,T,XY,W                                 BMD50D0266
000012        DIMENSION TP(1),TC(1)                                              BMD50D0267
000012        COMMON V,CLAB,XY,SYM,Y                                             BMD50D0268
000012        COMMON /PRESET/ NCC                                                BMD50D0269
000012        DATA (NCC = 0)                                                     BMD50D0270
000012   100    FORMAT(1H 8A5(F12.3,8X),F12.3/17A,5(F12.3,9X))                   BMD50D0271
000012   101  FORMAT(1H F12.3,1X,A1,16A6,A5,A1,F12.3)                            BMD50D0272
000012    102  FORMAT(1H 13X,A1,16A6,A5,A1)                                      BMD50D0273
000012   1000  FORMAT(1H  14X,101A1)                                              BMD50D0274
000012   1001  FORMAT(15X,20(5H*,***),1H*)                                        BMD50D0275
000012        BLANKS=(*6H      )                                                 BMD50D0276
000013        IF(NCC)48,50,48                                                    BMD50D0277
000014     50  KL=0                                                              BMD50D0278
000015        GF(1)=(*6H1X    )                                                  BMD50D0279
000017        GF(2)=(*6H2X    )                                                  BMD50D0280
000020        GF(3)=(*6H3X    )                                                  BMD50D0281
000022        GF(4)=(*6H4X    )                                                  BMD50D0282
000023        GF(5)=(*6H5X    )                                                  BMD50D0283
000025        GF(6)=(*6H6X    )                                                  BMD50D0284
000026        GF(7)=(*6H7X    )                                                  BMD50D0285
000030        GF(8)=(*6H8X    )                                                  BMD50D0286
000031        GF(9)=(*6H9X    )                                                  BMD50D0287
000033        GF(10)=(*6H10X   )                                                 BMD50D0288
000034        FMT(1)=(*6H(17X  )                                                 BMD50D0289
000036        FMT(2)=BLANKS                                                      BMD50D0290
000037        FMT(3)=BLANKS                                                      BMD50D0291
000040        FMT(4)=(*6H5(F12.)                                                 BMD50D0292
000041        FMT(5)=(*6H3,8X)/)                                                 BMD50D0293
000043        FMT(6)=(*6H7X,   )                                                 BMD50D0294
000044        FMT(8)=(*6H4(F12.)                                                 BMD50D0295
000046        FMT(9)=(*6H3,8X),)                                                 BMD50D0296
000047        FMT(10)=(*6HF12.3))                                                BMD50D0297
000051        TC=(*1H,)                                                          BMD50D0298
000052        TP=(*1H*)                                                          BMD50D0299
000054        CALL SCALE(WMIN,WMAX,100,0,JY,YMIN,YMAX,YI,0)                      BMD50D0300
```

```
000062          YR=YMAX-YMIN                                                    BMD5D00301
000064      230 J=JY                                                            BMD5D00302
000066          IF(J*(J-10))204,201,201                                         BMD5D00303
000075      201 IF(KL)220,220,231                                               BMD5D00304
000077      231 WRITE(            6,1001)                                       BMD5D00305
000103          IF(KL)250,250,220                                               BMD5D00306
000111      220 CLAB(1)= YMIN                                                   BMD5D00307
000113          DO 222  I=2,11                                                  BMD5D00308
000114      222 CLAB(I)=CLAB(I-1)+YIJ                                           BMD5D00309
000122          WRITE(            6,100)(CLAB(I),I=1,11,2),(CLAB(J),J=2,10,2)   BMD5D00310
000141          IF(KL)231,231,14                                                BMD5D00311
000147      204 IF(J-5)205,221,207                                              BMD5D00312
000152      207 J=J-5                                                           BMD5D00313
000154      205 JYT=5*J                                                         BMD5D00314
000156      221 CONTINUE                                                        BMD5D00315
000156          IF(KL)226,226,227                                               BMD5D00316
000160      226 FMT(3)=GF(JY)                                                   BMD5D00317
000163      225 FMT(7)=GF(JY)                                                   BMD5D00318
000166          TT=JY                                                           BMD5D00319
000167          TT=TT*YIJ/10.                                                   BMD5D00320
000171          CLAB(1)=YMIN+TT                                                 BMD5D00321
000173          DO 223 I=2,10                                                   BMD5D00322
000175      223 CLAB(I)=CLAB(I-1) +YIJ                                          BMD5D00323
000203          WRITE(            6,FMT)(CLAB(I),I=2,10,2),(CLAB(J),J=1,9 ,2)  BMD5D00324
000223          IF(KL)227,227,14                                                BMD5D00325
000231      227 IF(JY-5)208,209,208                                             BMD5D00326
000233      209 WRITE(            6,1002)                                       BMD5D00327
000237          IF(KL)250,250,226                                               BMD5D00328
000245      208 J = 5 - JYT                                                     BMD5D00329
000247          WRITE(            6,1000)(TC,I=1,J ),((IP,(IC,I=1,4)),K=1,19),IP,  BMD5D00330
               1 (IC,I=1,JYT)                                                   BMD5D00331
000306          IF(KL)250,250,226                                               BMD5D00332
000314      250 CONTINUE                                                        BMD5D00333
000314          NCC=1                                                           BMD5D00334
000315          IC=0                                                            BMD5D00335
000316          IF(NP)80,11,11                                                  BMD5D00336
000320       11 DO 1 I=1,51                                                     BMD5D00337
000322          DO 1 J=1,17                                                     BMD5D00338
000323       1  XY(I,J)=BLANKS                                                  BMD5D00339
000333          CALL SCALE (ZMIN,ZMAX,50.,JX,XMIN,XMAX,XIJ)                     BMD5D00340
000341          XR=XMAX-XMIN                                                    BMD5D00341
000343       48 IF(NC)52,13,49                                                  BMD5D00342
000350       49 IF(NP)80,10,10                                                  BMD5D00343
000352       10 DO 9 N=1,NC                                                     BMD5D00344
000354          SYMH = SYM(N)                                                   BMD5D00345
000356           XDIFFR=XMAX-X                                                  BMD5D00346
000360           IF(XDIFFR)105,106,106                                          BMD5D00347
000361      105  XDIFFR=0.0                                                     BMD5D00348
000362      106  YDIFFR=YMAX-Y(N)                                               BMD5D00349
000366           IF(YDIFFR)107,108,108                                          BMD5D00350
000367      107  YDIFFR=0.0                                                     BMD5D00351
000370      108  L=51.-(50.*XDIFFR)/XR+.5                                       BMD5D00352
000377           K=101.-(100.*YDIFFR)/YR+.5                                     BMD5D00353
000405          M= MOD (K,6)                                                    BMD5D00354
000411           K=(K-1)/6+1                                                    BMD5D00355
000415          IF(M)21,16,21                                                   BMD5D00356
000416       16 M=6                                                             BMD5D00357
000417       21 LL=M                                                            BMD5D00358
000421              M=(M-1)*6                                                   BMD5D00359
          C                                                                     BMD5D00360
          C     BATTELLE COMMENT                                              BMD5D00361
          C     MASK IN PROPER CHARACTER - SUBROUTINE ADDED BY BATTELLE TO     BMD5D00362
          C     REPLACE INDIANA CO-OP ROUTINE CALLED FORM2.                   BMD5D00363
```

```
C                                                                       BMD5D00364
000423          CALL FORM3(XY(L,K), LL, SYMB)                           BMD5D00365
C                                                                       BMD5D00366
000432    9      CONTINUE                                               BMD5D00367
000440           GO TO 15                                               BMD5D00368
000440       80 DO 86 I=1,17                                            BMD5D00369
000442       86  XY(1,I)=BLANKS                                         BMD5D00370
000450           L=1                                                    BMD5D00371
000451           DO 95 N=1,NC                                           BMD5D00372
000452           SYMB=SYM(N)                                            BMD5D00373
000454            YDIFFR=YMAX-Y(N)                                      BMD5D00374
000457            IF(YDIFFR)860,865,865                                 BMD5D00375
000460       860  YDIFFR=0.0                                            BMD5D00376
000461       865  K=101.-(100.*YDIFFR)/YR+.5                            BMD5D00377
000470           M= MOD (K,6)                                           BMD5D00378
000474           IF(M)90,91,90                                          BMD5D00379
000475        91 M=6                                                    BMD5D00380
000476        90 LL=M                                                   BMD5D00381
000500            K=(K-1)/6+1                                           BMD5D00382
000504               M=(M-1)*5                                          BMD5D00383
C                                                                       BMD5D00384
C       BATTELLE COMMENT                                                BMD5D00385
C       MASK IN PROPER CHARACTER - SUBROUTINE ADDED BY BATTELLE TO      BMD5D00386
C       REPLACE INDIANA CO-OP ROUTINE CALLED FORM2.                     BMD5D00387
C                                                                       BMD5D00388
000506    95   CALL FORM3(XY(L,K), LL, SYMB)                           BMD5D00389
C                                                                       BMD5D00390
000523           IF( MOD (IC,5))97,96,97                                BMD5D00391
000527        96  W=TP                                                  BMD5D00392
000531           GO TO 98                                               BMD5D00393
000531        97  W=TC                                                  BMD5D00394
000533        98 WRITE(             6,101)X,W,(XY(1,N),N=1,17),W,X      BMD5D00395
000562           IC=IC+1                                                BMD5D00396
000564            GO TO 15                                              BMD5D00397
000570        13  M=6-JX                                               BMD5D00398
000572            LL=50+M                                               BMD5D00399
000574            T=JX                                                  BMD5D00400
000575            IF(5-JX)131,131,135                                   BMD5D00401
000577       131  T=0.0                                                 BMD5D00402
000600       135  RLAB=XMAX-(T*XIJ)/5.0                                BMD5D00403
000605            W=TC                                                  BMD5D00404
000606            K=52                                                  BMD5D00405
000607            DO 31 L=M,LL                                          BMD5D00406
000611            K=K-1                                                 BMD5D00407
000613            I= MOD (L,5)                                          BMD5D00408
000617            IF(I-1)2,3,2                                          BMD5D00409
000621         3  W=TP                                                  BMD5D00410
000623            WRITE(             6,101)RLAB,W,(XY(K,N),N=1,17),W,RLAB BMD5D00411
000646            RLAB=RLAB-XIJ                                         BMD5D00412
000650            W=TC                                                  BMD5D00413
000652            GO TO 31                                              BMD5D00414
000656         2  WRITE(             6,102)W,(XY(K,N),N=1,17),W        BMD5D00415
000676        31  CONTINUE                                              BMD5D00416
000704        52  KL=1                                                  BMD5D00417
000705            GO TO 230                                             BMD5D00418
000706        14  NCC=0                                                 BMD5D00419
000707        15  RETURN                                                BMD5D00420
000710                END                                               BMD5D00421

SUBPROGRAM LENGTH
001164
```

```
PLOTR

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1       - 000323    2       - 000656    3       - 000621    10      - 000352
11      - 000320    13      - 000570    14      - 000706    15      - 000707
18      - 000416    21      - 000417    31      - 000676    48      - 000343
49      - 000350    50      - 000014    52      - 000704    80      - 000440
86      - 000442    90      - 000476    91      - 000475    95      - 000527
97      - 000531    98      - 000533    100     - 000713    101     - 000721
102     - 000726    105     - 000361    106     - 000362    107     - 000367
108     - 000370    131     - 000577    135     - 000600    201     - 000075
204     - 000147    205     - 000154    207     - 000152    208     - 000245
209     - 000233    220     - 000111    241     - 000156    222     - 000115
223     - 000176    225     - 000163    226     - 000160    227     - 000231
230     - 000064    231     - 000077    250     - 000314    880     - 000460
865     - 000461    1000    - 000732    1001    - 000735

BLOCK NAMES AND LENGTHS
        - 037122    PRESET  - 000001

VARIABLE ASSIGNMENTS
BLANKS  - 001126    CLAB    - 035230C01 FMT     - 001114    GF      - 001102
I       - 001142    IC      - 001146    J       - 001141    JX      - 001147
JY      - 001134    JYT     - 001143    K       - 001145    KL      - 001133
L       - 001160    LL      - 001142    M       - 001161    N       - 001154
NCC     - 000000C02 NP      - 000000    RLAB    - 001163    SYM     - 037064C01
SYMB    - 001155    T       - 001131    TC      - 001127    TD      - 001130
TT      - 001144    V       - 000000C01 W       - 001132    XDIFFR  - 001156
XIJ     - 001152    XM      - 001074    XMAX    - 001151    XMIN    - 001150
XR      - 001153    XY      - 035321C01 Y       - 037103C01 YDIFFR  - 001157
YIJ     - 001137    YMAX    - 001136    YMIN    - 001135    YR      - 001140

START OF CONSTANTS
000712

START OF TEMPORARIES
001022

START OF INDIRECTS
001072

UNUSED COMPILER SPACE
017400

                SUBROUTINE TRANS(N,NJ,NTR)                                  BMD5D00422
      CTRANS          SUBROUTINE TRANS FOR BMD05D          JULY 17, 1964    BMD5D00423
000006          DIMENSION DATA (15000)                                      BMD5D00424
000006          COMMON DATA                                                 BMD5D00425
000006          TYPE INTEGER C123,TODE                                      BMD5D00426
000006          ASNF(X)=ATAN (X/SQRT (1.0-X**2))                            BMD5D00427
                C                                                           BMD5D00428
000023          C123=(*6HTRNGEN)                                            BMD5D00429
000024          ON=N+1                                                      BMD5D00430
000026          MARY=0                                                      BMD5D00431
000030          WRITE(              6,1403)                                 BMD5D00432
000033          WRITE(              6,1400)                                 BMD5D00433
000037          IERROR=0                                                    BMD5D00434
000040          DO 1000 I=1,NTR                                             BMD5D00435
000044          READ(           5,900)TODE,NE,NC,NV,CC                      BMD5D00436
000061          IF (TODE-C123) 300,6,300                                    BMD5D00437
```

```
000065      300 NJ=-NJ                                                          BMD5D00438
000066          RETURN                                                          BMD5D00439
000067        6 WRITE(              6,1402)I,NE,NC,NV,CO                        BMD5D00440
000105          MA=N*NE-N                                                       BMD5D00441
000111          MB=N*NV-N+1                                                     BMD5D00442
000115          MC=MB+N-1                                                       BMD5D00443
000116          IF(NC*(15-NC))1500,1500,2                                       BMD5D00444
000122     1500 WRITE(              6,1406)                                     BMD5D00445
000126          GO TO 1000                                                      BMD5D00446
000131     2    IF(NC-11) 4, 3, 3                                               BMD5D00447
000134        3 K=CO                                                            BMD5D00448
000136          MD=N*K-N                                                        BMD5D00449
000141        4 DO 200 J=MB,MC                                                  BMD5D00450
000143          MA=MA+1                                                         BMD5D00451
000145          MD=MD+1                                                         BMD5D00452
000146        5 CONTINUE                                                        BMD5D00453
000146          GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140),NC       BMD5D00454
000170       10 IF(DATA(J))99,32,8                                              BMD5D00455
000173        8 DATA(MA)=SQRT (DATA(J))                                         BMD5D00456
000201          GO TO 200                                                       BMD5D00457
000203       20 IF(DATA(J))99,11,12                                             BMD5D00458
000206       11 DATA(MA)=1.0                                                    BMD5D00459
000211          GO TO 200                                                       BMD5D00460
000211       12 DATA(MA)=SQRT (DATA(J))+SQRT (DATA(J)+1.0)                      BMD5D00461
000226          GO TO 200                                                       BMD5D00462
000230       30 IF(DATA(J))99,99,14                                             BMD5D00463
000233       14 DATA(MA)=ALOG(DATA(J))*0.43429448l9                             BMD5D00464
000241          GO TO 200                                                       BMD5D00465
000244       40 DATA(MA)=EXP (DATA(J))                                          BMD5D00466
000252          GO TO 200                                                       BMD5D00467
000254       50 IF(DATA(J))99,32,17                                             BMD5D00468
000257       17 IF(DATA(J)-1.0)18,19,99                                         BMD5D00469
000263       19 DATA(MA)=3.1415926536/2.0                                       BMD5D00470
000266          GO TO 200                                                       BMD5D00471
000266       18 A=SQRT (DATA(J))                                                BMD5D00472
000272          DATA(MA)=ASNF(A)                                                BMD5D00473
000300          GO TO 200                                                       BMD5D00474
000300       60 A=DATA(J)/ON                                                    BMD5D00475
000303          B=A+1.0/ON                                                      BMD5D00476
000305          IF(A)99,23,24                                                   BMD5D00477
000307       23 IF(B)99,26,27                                                   BMD5D00478
000311       26 DATA(MA)=0.0                                                    BMD5D00479
000313          GO TO 200                                                       BMD5D00480
000314       27 DATA(MA)=ASNF(SQRT (B))                                         BMD5D00481
000325          GO TO 200                                                       BMD5D00482
000325       24 IF(B)99,28,29                                                   BMD5D00483
000327       28 DATA(MA)=ASNF(SQRT (A))                                         BMD5D00484
000340          GO TO 200                                                       BMD5D00485
000340       29 A=SQRT (A)                                                      BMD5D00486
000342          B=SQRT (B)                                                      BMD5D00487
000345          DATA(MA)=ASNF(A)+ASNF(B)                                        BMD5D00488
000357          GO TO 200                                                       BMD5D00489
000357       70 IF(DATA(J))31,99,31                                             BMD5D00490
000361       31 DATA(MA)=1.0/DATA(J)                                            BMD5D00491
000365          GO TO 200                                                       BMD5D00492
000366       80 DATA(MA)=DATA(J)+CO                                             BMD5D00493
000373          GO TO 200                                                       BMD5D00494
000373       90 DATA(MA)=DATA(J)*CO                                             BMD5D00495
000377          GO TO 200                                                       BMD5D00496
000400      100 IF(DATA(J))33,32,33                                             BMD5D00497
000402       32 DATA(MA)=0.0                                                    BMD5D00498
000404          GO TO 200                                                       BMD5D00499
```

```
000435       33 DATA(MA)=DATA(J)**C0                                          BMD5D00500
000413          GO TO 200                                                     BMD5D00501
000414      110 DATA(MA)=DATA(J)+DATA(MD)                                     BMD5D00502
000422          GO TO 200                                                     BMD5D00503
000422      120 DATA(MA)=DATA(J)-DATA(MD)                                     BMD5D00504
000430          GO TO 200                                                     BMD5D00505
000430      130 DATA(MA)=DATA(J)*DATA(MD)                                     BMD5D00506
000435          GO TO 200                                                     BMD5D00507
000436      140 IF(DATA(MD))157,99,157                                        BMD5D00508
000440      157 DATA(MA)=DATA(J)/DATA(MD)                                     BMD5D00509
000445          GO TO 200                                                     BMD5D00510
000446       99 IF(MARY)43,44,44                                              BMD5D00511
000450       44 MARY=-999                                                     BMD5D00512
000451          IERROR=-999                                                   BMD5D00513
000452          WRITE(            6,1404)I                                    BMD5D00514
000460       43 WRITE(            6,1405)J                                    BMD5D00515
000466      200 CONTINUE                                                      BMD5D00516
000473          MARY=0                                                        BMD5D00517
000474     1000 CONTINUE                                                      BMD5D00518
000477          IF(IERROR)42,1111,1111                                        BMD5D00519
000500       42 WRITE(            6,1401)                                     BMD5D00520
000504     1111 RETURN                                                        BMD5D00521
          C                                                                 BMD5D00522
000505      900 FORMAT(A6,I3,I2,I3,F6.0)                                      BMD5D00523
          C                                                                 BMD5D00524
000505     1400 FORMAT(46H0CARD    NEW     TRANS    ORIG.   ORIG. VAR(B)/45H  NO.  BMD5D00525
              1VARIABLE   CODE    VAR(A)   OR CONSTANT)                     BMD5D00526
000505     1401 FORMAT(78H VALUES OF VARIABLES OF WHICH AN ERROR WAS FOUND DURING  BMD5D00527
              1TRANS-GENERATION WILL /77H STILL BE INCLUDED IN THE GRAPHS. HOWEVE  BMD5D00528
              2R, THESE GRAPHS MAY BE MEANINGLESS  /54H SINCE SOME VALUES WILL B  BMD5D00529
              3E TRANSFORMED AND OTHERS NOT.)                               BMD5D00530
000505     1402 FORMAT(2H  I2,I8,2I9,4X,F10.5)                               BMD5D00531
000505     1403 FORMAT(1H06X,23HTRANS GENERATOR CARD(S))                     BMD5D00532
000505     1404 FORMAT(55H0THE INSTRUCTIONS INDICATED ON TRANS GENERATOR CARD NO.I  BMD5D00533
              12,1X,3HRE-/60H SULTED IN THE VIOLATION OF A RESTRICTION FOR THIS T  BMD5D00534
              2RANSFOR-/59H MATION. THE VIOLATION OCCURRED FOR THE ITEMS LISTED B  BMD5D00535
              3ELOW.)                                                       BMD5D00536
000505     1405 FORMAT(10H ITEM NO. I5)                                      BMD5D00537
000505     1406 FORMAT(107H0TRANSGENERATION CODE ON CARD LISTED ABOVE IS INCORRECT  BMD5D00538
              X. PROGRAM WILL PROCEED WITHOUT THIS TRANSGENERATION.)        BMD5D00539
          C                                                                 BMD5D00540
000505          END                                                         BMD5D00541
```

SUBPROGRAM LENGTH
000732

TRANS

FUNCTION ASSIGNMENTS
ASNF = 000010

STATEMENT ASSIGNMENTS

2	=	000131	3	=	000134	4	=	000141	5	=	000146
6	=	000067	8	=	000173	10	=	000170	11	=	000206
12	=	000211	14	=	000233	17	=	000257	18	=	000266
19	=	000263	20	=	000203	23	=	000307	24	=	000325
26	=	000311	27	=	000314	28	=	000327	29	=	000340
30	=	000230	31	=	000361	32	=	000402	33	=	000405
40	=	000244	42	=	000500	43	=	000460	44	=	000450
50	=	000254	60	=	000300	70	=	000357	80	=	000366
90	=	000373	99	=	000446	100	=	000400	110	=	000414
120	=	000422	130	=	000430	140	=	000436	157	=	000440
200	=	000466	300	=	000065	900	=	000521	1000	=	000474
1111	=	000504	1400	=	000524	1401	=	000540	1402	=	000570
1403	=	000574	1404	=	000601	1405	=	000626	1406	=	000631
1500	=	000122									

```
BLOCK NAMES AND LENGTHS
       = 035230

VARIABLE ASSIGNMENTS
A       = 000730    B       = 000731    CO      = 000721    C1?J    = 000710
DATA    = 000000C01 I       = 000715    IERROR  = 000714    J       = 000727
K       = 000725    MA      = 000722    MARY    = 000713    MB      = 000723
MC      = 000724    MD      = 000726    NC      = 000717    NE      = 000716
NV      = 000720    ON      = 000712    TUDE    = 000711

START OF CONSTANTS
000507

START OF TEMPORARIES
000646

START OF INDIRECTS
000702

UNUSED COMPILER SPACE
020400



              SUBROUTINE VFCHCK(NVF)                                          BMD5D00542
         CVFCHCK    SUBROUTINE TO CHECK FOR PROPER NUMBER OF VARIABLE FORMAT CRDS BMD5D00543
000003           IF(NVF)10,10,20                                              BMD5D00544
000006    10     WRITE(            6,4000)                                    BMD5D00545
000010           NVF=1                                                        BMD5D00546
000012    50     RETURN                                                       BMD5D00547
         C                                                                   BMD5D00548
000013    20     IF(NVF-10)50,50,10                                           BMD5D00549
         C                                                                   BMD5D00550
000016    4000 FORMAT(1H023X71HNUMBER OF VARIABLE FORMAT CARDS INCORRECTLY SPECIF BMD5D00551
             KIED, ASSUMED TO BE 1.)                                         BMD5D00552
000016           END                                                          BMD5D00553

SUBPROGRAM LENGTH
000036
VFCHCK

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10      = 000004    20      = 000013    50      = 000012    4000    = 000022

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

START OF CONSTANTS
000020

START OF TEMPORARIES
000034

START OF INDIRECTS
000036

UNUSED COMPILER SPACE
022600
```

```
                SUBROUTINE SCALE(YMIN,YMAX,YINT,JY,TYMIN,TYMAX,YIJ)                    BMD5D00554
          CSCALE          SUBROUTINE SCALE FOR SUB PLOTR          AUGUST 18, 1964      BMD5D00555
000012            DIMENSION C(10)                                                      BMD5D00556
000012             C(1)=  1.0                                                          BMD5D00557
000013             C(2)=1.5                                                            BMD5D00558
000014             C(3)=2.0                                                            BMD5D00559
000016             C(4)=3.0                                                            BMD5D00560
000017             C(5)=4.0                                                            BMD5D00561
000021           C(6)=5.0                                                              BMD5D00562
000022             C(7)=7.5                                                            BMD5D00563
000024             C(8)=10.0                                                           BMD5D00564
000025           TEST=.5*(2**(-24))                                                    BMD5D00565
000033       50  YR=YMAX-YMIN                                                         BMD5D00566
000034           TT=YR/YINT                                                          BMD5D00567
000036           J=ALOG(TT)*0.43429448819                                            BMD5D00568
000042            E=10.0**J                                                          BMD5D00569
000046            TT=TT/E                                                            BMD5D00570
000047            I=0                                                                BMD5D00571
000050            IF(TT-1.0)205,201,201                                              BMD5D00572
000056       205  TT=TT*10.0                                                         BMD5D00573
000060            E=E/10.0                                                           BMD5D00574
000062       201  I=I+1                                                              BMD5D00575
000064            IF(8-I)1,2,2                                                       BMD5D00576
000066         1  E=E*10.0                                                          BMD5D00577
000070            I=1                                                                BMD5D00578
000071         2  IF(TT-C(I))233,202,201                                            BMD5D00579
000075       233  YIJ=C(I)*E                                                         BMD5D00580
000101            GO TO 203                                                          BMD5D00581
000102       202  Y=YMIN/C(I)                                                       BMD5D00582
000105            J=Y                                                                BMD5D00583
000107            T=J                                                                BMD5D00584
000110            IF(0.0001-ABS (T-Y))204,233,233                                    BMD5D00585
000115       204  YIJ=C(I+1)*E                                                       BMD5D00586
000120       203  X=((YMAX+YMIN)/YIJ-YINT )/2.0+.00001                              BMD5D00587
000125            K=X                                                                BMD5D00588
000127            IF(K)235,240,240                                                   BMD5D00589
000131       235  Y=K                                                                BMD5D00590
000132            IF(X-Y)236,240,236                                                 BMD5D00591
000135       236  K=K-1                                                              BMD5D00592
000137       240  TYMIN=K                                                            BMD5D00593
000140            TYMIN=YIJ*TYMIN                                                    BMD5D00594
000142           TYMAX=TYMIN+YINT*YIJ                                                BMD5D00595
000144            IF(YMAX-TYMAX-TEST)10,10,201                                       BMD5D00596
000147        10  TT=YINT/10.                                                       BMD5D00597
000150            JY=TT+.000001                                                      BMD5D00598
000153            YIJ=YINT*(YIJ/10.0)                                                BMD5D00599
000155            J=TYMIN/ YIJ                                                       BMD5D00600
000157            IF (K)242,241,241                                                  BMD5D00601
000161       242  J=J-1                                                             BMD5D00602
000163       241  J=J*JY+JY-K                                                       BMD5D00603
000167            JY=J                                                               BMD5D00604
000170            RETURN                                                             BMD5D00605
000176            END                                                                BMD5D00606

SUBPROGRAM LENGTH
000244
SCALE

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1     = 000066   2     = 000071   10    = 000147   50    = 000033
201   = 000062   202   = 000102   203   = 000120   204   = 000115
205   = 000056   233   = 000075   235   = 000131   236   = 000135
240   = 000137   241   = 000163   242   = 000161
```

```
BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
C       =  000220    E       =  000236    I       =  000237    J       =  000235
K       =  000243    T       =  000241    TEST    =  000232    TT      =  000234
X       =  000242    Y       =  000240    YIJ     =  000000    YR      =  000233

START OF CONSTANTS
000172

START OF TEMPORARIES
000211

START OF INDIRECTS
000217

UNUSED COMPILER SPACE
021700


                SUBROUTINE FILL(SYM, NC)                                        BMD5D00607
          C                                                                     BMD5D00608
          C     BATTELLE SUBROUTINE                                             BMD5D00609
          C                                                                     BMD5D00610
          C     THIS ROUTINE FILLS THE COMPUTER WORD WITH THE INPUT SYMBOL SYM(N)  BMD5D00611
          C     ADDED TO INDIANA CO-OP VERSION IN ORDER TO COMPLIMENT ADDED     BMD5D00612
          C     BATTELLE SUBROUTINE CALLED FORM3.                               BMD5D00613
          C                                                                     BMD5D00614
          C     THIS VERSION IS FOR A X-BYTE WORD, WHERE X = 10.                BMD5D00615
          C                                                                     BMD5D00616
000005            DIMENSION SYM(15)                                             BMD5D00617
          C                                                                     BMD5D00618
000005            DO 10 N=1,NC                                                  BMD5D00619
000006     10     ENCODE(10,1000,SYM(N)) (SYM(N), I=1,10)                       BMD5D00620
000031            RETURN                                                        BMD5D00621
          C                                                                     BMD5D00622
000031     1000   FORMAT(10R1)                                                  BMD5D00623
000031            END                                                           BMD5D00624

SUBPROGRAM LENGTH
000043
FILL

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1000    =  000034

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
I       =  000042    N       =  000041

START OF CONSTANTS
000043

START OF TEMPORARIES
000036

START OF INDIRECTS
000041

UNUSED COMPILER SPACE
022500
```

```
                SUBROUTINE FORM3(GRAPH, LOC, SYMBOL)                                    BMD5D00625
      C                                                                                 BMD5D00626
      C         BATTELLE SUBROUTINE                                                     BMD5D00627
      C                                                                                 BMD5D00628
      C         THIS SUBROUTINE PERFORMS ALL THE MASKING OPERATIONS FOR PLOTR.          BMD5D00629
      C                                                                                 BMD5D00630
      C         GRAPH = THE COMPUTER WORD AT THE COORDINATES TO BE LOADED.              BMD5D00631
      C         LOC = THE BYTE LOCATION WITHIN THE COMPUTER WORD FOR LOADING.           BMD5D00632
      C         SYMBOL = THE INPUT CHARACTER TO BE LOADED.  MUST FILL THE               BMD5D00633
      C         SIX LEFT BCD CHARACTERS OF THE COMPUTER WORD.                           BMD5D00634
      C                                                                                 BMD5D00635
      C         THIS ROUTINE AND THE ENTIRE PROGRAM TREATS THE PLOTTING WORD            BMD5D00636
      C         AS IF IT WERE ONLY 6 ALPHANUMERIC CHARACTERS LONG, LEFT-JUSTIFIED.      BMD5D00637
      C                                                                                 BMD5D00638
      C         THIS ROUTINE LOADS THE INPUT SYMBOL AT THE LOCATION LOC WITHIN          BMD5D00639
      C         THE COMPUTER WORD GRAPH.                                                BMD5D00640
      C         IF AN ITEM ALREADY EXISTS AT A LOCATION TO BE LOADED,                   BMD5D00641
      C         THE ROUTINE WILL LOAD AN NUMBER OR LETTER INDICATING THE NUMBER         BMD5D00642
      C         OF ITEMS AT THAT LOCATION.  REFER TO TABLE FOR MULTIPLE-ITEM            BMD5D00643
      C         ALPHANUMERICS.  THE NUMBER OF ITEMS, N, IS EQUAL TO THE                 BMD5D00644
      C         CHARACTER STORED AT TABLE(N-1).                                         BMD5D00645
      C         NOT ALL CALLING PROGRAMS USE THIS MULTIPLE-ITEM FEATURE, BUT            BMD5D00646
      C         IT HAS BEEN INCLUDED FOR THE SAKE OF STANDARDIZATION.                   BMD5D00647
      C                                                                                 BMD5D00648
000006          DIMENSION TABLE(18), MASK(9)                                            BMD5D00649
000006          INTEGER TABLE, GRAPH, SYMBOL, BLANKS                                    BMD5D00650
000006          DATA(BLANKS=6H      )                                                   BMD5D00651
      C                                                                                 BMD5D00652
000006          DATA(TABLE=6H222222,6H333333,6H444444,6H555555,6H666666,6H777777,       BMD5D00653
     1                   6H888888,6H999999,6HAAAAAA,6HBBBBBB,6HCCCCCC,6HDDDDDD,       BMD5D00654
     2                   6HEEEEEE,6HFFFFFF,6HGGGGGG,6HHHHHH,6HIIIIII,6H//////)       BMD5D00655
      C                                                                                 BMD5D00656
000006          DATA (MASK = 77000000000000000000B, 00770000000000000000B,             BMD5D00657
     1                       00007700000000000000B, 00000077000000000000B,             BMD5D00658
     2                       00000000770000000000B, 00000000007700000000B)             BMD5D00659
      C                                                                                 BMD5D00660
      C                                                                                 BMD5D00661
      C         MASK OUT LOCATION UNDER INVESTIGATION.                                  BMD5D00662
000006          ITEM = GRAPH.AND.MASK(LOC)                                              BMD5D00663
      C                                                                                 BMD5D00664
      C         TEST TO SEE IF THERE ARE ALREADY 19 OR MORE ITEMS AT LOCATION         BMD5D00665
000010          LOGIC = TABLE(18).AND.MASK(LOC)                                         BMD5D00666
000012          IF(ITEM.EQ.LOGIC) RETURN                                                BMD5D00667
      C                                                                                 BMD5D00668
      C         TEST TO SEE IF BLANK (IE, NO PREVIOUS ITEMS AT LOCATION).               BMD5D00669
000015          LOGIC = BLANKS.AND.MASK(LOC)                                            BMD5D00670
000020          IF(ITEM.NE.LOGIC) GO TO 10                                              BMD5D00671
      C                                                                                 BMD5D00672
      C         IF BLANK AT LOCATION, MASK IN SYMBOL.                                   BMD5D00673
000022          GRAPH = (GRAPH.AND.(.NOT.MASK(LOC))).OR.(SYMBOL.AND.MASK(LOC))          BMD5D00674
000027          RETURN                                                                  BMD5D00675
      C                                                                                 BMD5D00676
      C         SCAN THRU MULTIPLE-ITEM TABLE FOR MATCH.                                BMD5D00677
000030    10    DO 20 I=1,17                                                            BMD5D00678
000032          LOGIC = TABLE(I).AND.MASK(LOC)                                          BMD5D00679
000035          IF(ITEM.NE.LOGIC) GO TO 20                                              BMD5D00680
      C                                                                                 BMD5D00681
      C         IF MULTIPLE-SYMBOL IS FOUND, MASK IN NEXT SYMBOL.                       BMD5D00682
000040          GRAPH = (GRAPH.AND.(.NOT.MASK(LOC))).OR.(TABLE(I+1).AND.MASK(LOC))      BMD5D00683
000045          RETURN                                                                  BMD5D00684
      C                                                                                 BMD5D00685
000046    20    CONTINUE                                                                BMD5D00686
```

```
      C     IF NO MATCH IS FOUND, THEN MASK IN A 2.                          RMD5D00687
000050      GRAPH = (GRAPH.AND.(.NOT.MASK(LOC))).OR.(TABLE(I).AND.MASK(LOC))  RMD5D00688
000056      RETURN                                                           RMD5D00689
000056      END                                                              RMD5D00690

SUBPROGRAM LENGTH
000131




FORM3

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10     = 000030    20     = 000046

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
BLANKS = 000125    I      = 000130    ITEM   = 000126    LOGIC  = 000127
MASK   = 000117    TABLE  = 000075

START OF CONSTANTS
000060

START OF TEMPORARIES
000061

START OF INDIRECTS
000071

UNUSED COMPILER SPACE
022300
```

SCATTERGRAM ROUTINE

**for non-blank entries for both variables
in the same data set**

```
                PROGRAM SCAT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE8)
000003          DATA SC/1H /
000003          DIMENSION AA17(111),AA20(111)
000003          N = 0
000004          DO 100 K=1,2170
000006          READ(1,1) A20,A17
000015          READ(1,3) H
000023          READ(1,3) G
000031          READ(1,3) D
000037          READ(1,3) F
000045          READ(1,3) E
000053          READ(1,3) G
000061          READ(1,3) H
000067        1 FORMAT(3HA,A4,2X,A3)
000067        3 FORMAT(30A,85)
000067          IF((A17.NE.HL).AND.(A20.NE.HL)) GO TO 10
000077          GO TO 100
000077       10 N = N + 1
000101          AA17(N) = A17
000103          AA20(N) = A20
000104      100 CONTINUE
000106          WRITE(6,50) ((AA17(K),AA20(K)),K=1,N)
000123       50 FORMAT(2A4,2X,A4)
000123          END FILE 6
000125          END

PROGRAM LENGTH INCLUDING I/O BUFFERS
004651

UNUSED COMPILER SPACE
040200

77
        7
        8
        9
PROBLM  JULD00200111101000000    NO                                  0000801
(2X,F3.0,2X,F4.2)
PLOTSL00201001
FINISH
        6
        7
        8
        9
```

NOT REPRODUCIBLE

N.B. This insert indicates the deck set-up as submitted to the computer. These data cards for the BMD routine are not included in the output listing.

```
LOAD MAP      FILE = LGO                                   22.16.12. 12/19/69.   PAGE  1

FWA LOAD       100   LWA LOAD    12051   FWA LOADER  7E303   FWA TABLES  7E002      UNUSED STORAGE   63731

PROGRAM   ADDRESS   FILE                 COMMON    ADDRESS   LENGTH

SCAT         100    LGO
ENDFIL      4751    SYSTEM
INPUT       5044    SYSTEM
SYSIE       6165    SYSTEM               SCOPE2      6166
OUTPUT      7236    SYSTEM
FORTLIB    10576    SYSTEM
SIOP       11716    SYSTEM
SETAB      12032    SYSTEM

                                        /BLANK/        0         0
```

Continued on following page

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
SCAT	101								
ENDFIL	4752	SCAT	124						
INPUTC	5148	SCAT	7	11	13	16	17	21	22	25
			27	30	33	35	36	41	43	44
			47	51	52	55	57	60	63	65
			66							
KHAKER	5150									
QUNTRY	5166	SCAT	2							
SYSTEM	5373	ENDFIL	61							
		INPUTC	37	737						
		OUTPTC	16	1144						
SYSTEMC	5337									
SYSTEMD	5366									
END	5262	SCAT	125							
STOP	5312									
EXIT	5304									
ABNORML	5322	ENDFIL	62							
		INPUTC	40	730						
		OUTPTC	15	1145						
OUTPTC	7240	SCAT	110	114	116	122				
RUDER	7377									
FATAL7H	10577	SYSTEM	710							
CARD.F	10674	SIVS	56							
RESPRO.	11203									
FIZMAS.	11213									
POSFIL.	11241	ENDFIL	45							
		OUTPTC	32							
RDPRO.	11331									
DAT.	11352	OUTPTC	56	72	37					
CIOI.	11157	ENDFIL	37	53	15					
OPEN.	10716	ENDFIL	11							
		INPUTC	22							
		SYSTEM	506							
		OUTPTC	22							
SIO.	11035	INPUTC	52							
		OUTPTC	71							
ADVIN.	11252	ENDFIL	30							
		SYSTEM	410							
MVCS.	11064									

ENTRY	ADDRESS	REFERENCES (RELATIVE)			
POSFI.	11262	ENDFIL	24		
FIZMA.	11274				
DAT..	11502	INPUTC	55	94	97
BETHA	12032	ENDFIL	5		
		INPUTC	10		
		OUTPTC	10		

UNSATISFIED EXTERNALS REFERENCES (RELATIVE)

** NONE **

LOAD MAP FILE = LGO 22.23.16. 12/19/69. PAGE 1

FWA LOAD 100 FWA LOAD 96632 FWA LOADER 76303 FWA TABLES 74032 UNUSED STORAGE 5208

PROGRAM	ADDRESS	FILE	COMMON	ADDRESS	LENGTH
BMD2D	100	LGO			
COOL	11170	LGO			
CONV	11600	LGO			
PATTY	11712	LGO			
PLOT	12072	LGO			
TRNGEN	13230	LGO			
VECMOV	14025	LGO			
SCALE	14063	LGO			
ACSUER	14322	SYSTEM			
ALOG10	14334	SYSTEM			
ATAN	14423	SYSTEM			
ENDFIL	14517	SYSTEM			
EXP	14612	SYSTEM			
INPUTC	14671	SYSTEM			
SYSIE1	16012	SYSTEM	SCOPE2	16012	
INPUTB	17063	SYSTEM			
INPUTS	17406	SYSTEM			
OUTPTB	17470	SYSTEM			
OUTPTC	17725	SYSTEM			
OUTPTS	21265	SYSTEM			
RBALEA	21351	SYSTEM			
REWINM	21412	SYSTEM			
SQRT	21505	SYSTEM			
IMALEA	21550	SYSTEM			
RMAREA	21601	SYSTEM			
FATAL/B	21660	SYSTEM			
SIOS	21776	SYSTEM			
GETMA	23114	SYSTEM			
			/BLANK/	23133	43477

ENTRY	ADDRESS	REFERENCES (RELATIVE)					
BMD2D	101						
COOL	11200	BMD2D	562	573	647	660	674	705
CONV	11691	PLOT	517					
PATTY	11713	BMD2D	1205	1247	1314			
PLOT	12073	BMD2D	1323					
TRNGEN	13237	BMD2D	611	621	631	641	722	732
VECMOV	14026	BMD2D	202					
SCALE	14065	PLOT	46	61				
ACSUER	14323	BMD2D	551	577	609	711	1057	
		COOL	74	144				
		TRNGEN	74					
ALOG	14340	TRNGEN	144					
		SCALE	33					
		RMAREA	17					
ALOG10	14335							
ATAN	14424	TRNGEN	30					
ENDFIL	14520	BMD2D	1125					
EXP	14613	TRNGEN	175					
		RMAREA	22					

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
INPUTC	14673	BMU2D	35	40	42	44	46	50	52	54
			56	60	62	66	66	70	71	235
			237	241	243	245	247	250	323	327
			331	333	335	341	520	526	530	534
			541	545	1110	1120	1121			
		COVL	243	247	252					
		PLOT	175	201	204					
KRANEX	14775	INPUTS	5	23						
QNTRY	15013	BMU2D	2							
SYSTEM	15220	ACGDER	3							
		ALMLOG	40	42						
		ATAN	37							
		ENDFIL	61							
		EXP	33							
		INPUTC	37	727						
		INPUTB	41							
		INPUTS	26							
		OUTPTB	155							
		OUTPTC	14	1144						
		OUTPTS	46							
		RBAIEX	23							
		RE#INM	60							
		SQRT	25							
		THAIEX	17							
		XBANEX	37							
SYSTEMC	16164									
SYSTEMP	16213									
END	16107	BMU2D	1410							
		COVL	300							
		CONV	67							
		PAITY	111							
		PLOT	712							
		TRNGEN	447							
		VFCHCK	17							
		SCALE	165							
STOP	16137	BMU2D	156							
EXIT	16131	BMU2D	1406							
ABNORML	16147	ACGDER	4							
		ENDFIL	62							
		INPUTC	40	720						
		INPUTB	42							
		INPUTS	27							
		OUTPTB	156							
		OUTPTC	15	1148						
		OUTPTS	47							
		RE#INM	61							
INBUTB	17065	PLOT	361	363	364					
INBUTS	17410	BMU2D	75	77	101	103	105	107	111	113
			115	117	121	123	125	127	131	132
			403	405	407	410	415	417	421	422
			430	432	434	435				
		CONV	10	15	23	25				
		PLOT	145	150	154	157	212	214	218	220
			222	226	231					
OUTPTB	17472	BMU2D	755	757	760	1050	1052	1053		
OUTPOUT	17724									

ENTRY	ADDRESS	REFERENCES (RELATIVE)							
OUTPTC	41727	HMU2D	153	154	203	207	211	213	214	224
			225	230	231	255	257	261	263	265
			267	270	303	304	317	320	343	345
			353	355	357	361	363	364	1077	1141
			1102	1107	1110	1133	1134	1137	1143	1147
			1152	1159	1165	1171	1175	1202	1203	1213
			1214	1239	1250	1262	1264	1265	1272	1273
			1327	1330	1374	1376	1400	1401	1606	1645
		COUL	226	224	247					
		PAITY	40	44	50	55	57	66	72	
		PLUT	245	247	253	266	576	600	502	604
			614	622	630	633	641	644	554	662
			670	673						
		TRNGEN	422	424	429	430	432	433	435	447
		VFCHCK	6	7						
KOTER	41066	OUTPTS	5	23						
OUTPTS	41267	CONV	49	47	59	60				
		PLUT	107	111	113	115	117	121	123	125
			127	131	133	135	137	140		
RHAIEA	41352	SCALE	40							
REAINM	41413	HMU2D	134	1130						
		COUL	273							
		PLUT	165	516						
SORT	41506	HMU2D	1224							
		TRNGEN	25	199	143	151	216	244	261	276
			278							
INAIEA	41551	COUL	65							
RHAMEA	41602	TRNGEN	336							
FATALIB	41661	SYSTEM	710							
CARD.F	41756	SIOS	56							
ASSPRJ.	42265									
FIZMAT.	42275									
PUSFIL.	42323	ENUFIL	45							
		OUTPTC	32							
RUBRJ.	42413									
DAT.	42436	OUTPTC	56	79	87					
		OUTPTS	46	69	67					
CIOI.	42241	ENUFIL	37	69	15					
		INPUTH	70	74	103	25	202	239	267	
		OUTPTH	74	91	114	60	136	64	175	177
			221							
		REAINM	37	94	47					
OPEN.	42000	ENUFIL	11							
		INPUTC	22							
		SYSTEM	506							
		INPUTH	155	126						
		OUTPTH	75	131						
		OUTPTC	22							
SIO.	42117	INPUTC	52							
		INPUTH	34							
		OUTPTH	30							
		OUTPTC	71							
ADVIN.	42334	ENUFIL	30							
		SYSTEM	410							
		REAINM	34							
WRBUS.	42146	INPUTH	54							

ENTRY	ADDRESS	REFERENCES (RELATIVE)	
		OUTPTH	43
BOSFI.	42344	ENUFIL	24
		OUTPTH	126

Continued on following page

FIRST. 62350
DAT.. 62464

GET.. 63114

INPUTC	56	26	47	
INPUTS	17	33	42	45
ENDFIL	5			
INPUTC	10			
INPUTB	121			
OUTPTB	52			
OUTPTC	10			
REWIND	5			

UNSATISFIED EXTERNALS
REFERENCES (RELATIVE)
00 006 05

BMD02D CORRELATION WITH TRANSGENERATION - VERSION OF NOV. 13, 1964
HEALTH SCIENCES COMPUTING FACILITY, UCLA

PROBLEM CODE JB0D
NUMBER OF VARIABLES 2
NUMBER OF CASES 111

REMAINING SAMPLE SIZE= 111

SUMS

7180.0000 295.5100

MEANS

64.6847 2.6622

CROSS PRODUCT DEVIATIONS

	COL. 1	COL. 2
1	75687.9549	174.0457
2	174.0457	4.0691

STANDARD DEVIATIONS

26.2311 .1923

VARIANCE-COVARIANCE MATRIX

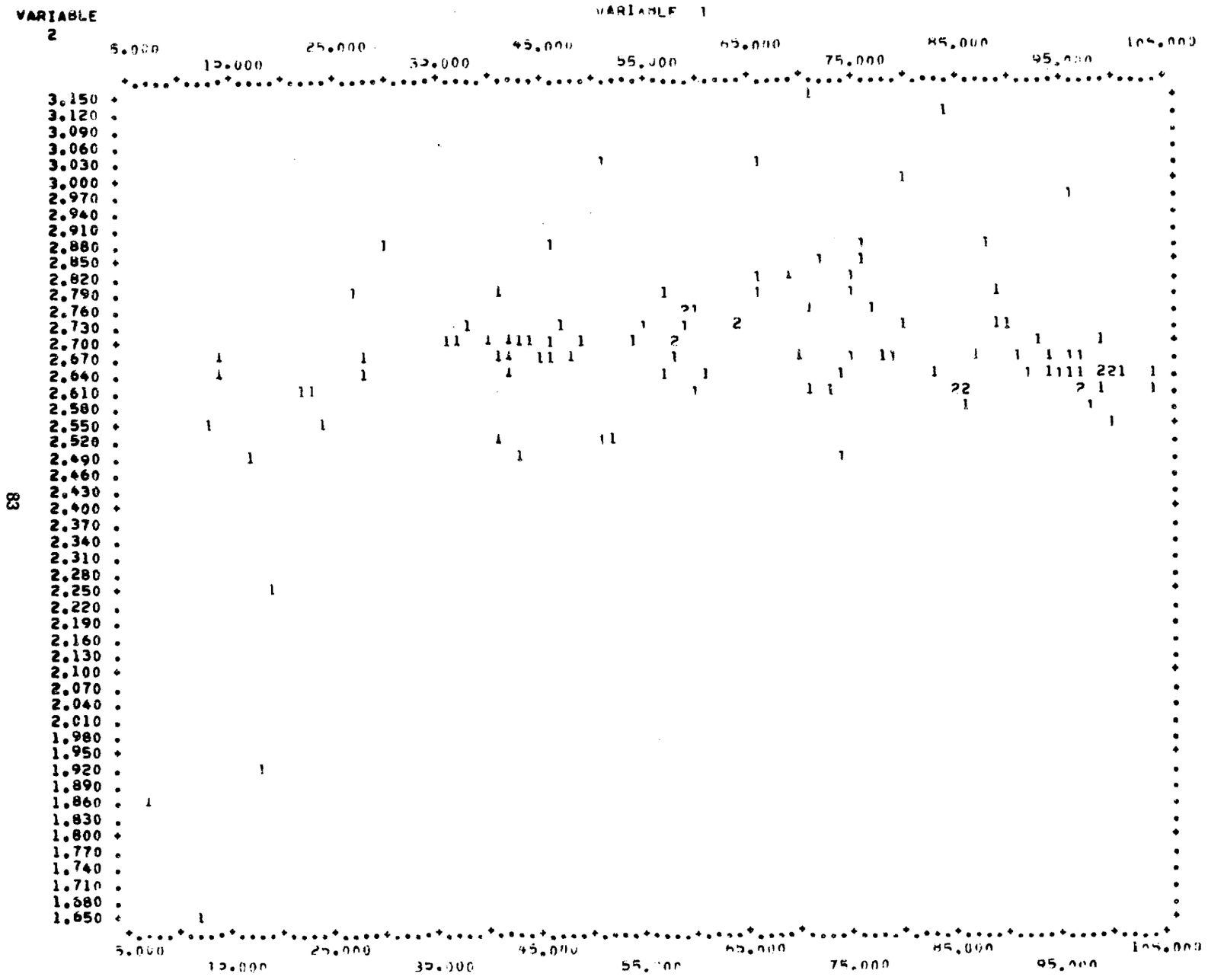
	COL. 1	COL. 2
1	688.0724	1.5822
2	1.5822	.0370

CORRELATION MATRIX

	COL. 1	COL. 2
1	1.0000	.3136
2	.3136	1.0000

```
MM53850.   12/19/69.[illegible] MAP 11/27/69.

22.05.24.MM53850/     351<,NAM=S,150,CM/60000,TP1,SIJ
22.05.24..
22.05.24.MAP(ON)
22.05.24.RUN(S)
22.05.28.CTIME 000.213 SEC.  R IN MOD LEVEL 05
22.05.29.REQUEST(TAPE1,555,HY,A,CER6,MT,WEAU)
22.16.09.  MT50 ASSIGNED = 555
22.16.10.REWIND(TAPE1)
22.16.10.LGO.
22.16.12.CX      .480 SEC.
22.16.12.PX     4.422 SEC.
22.16.12.NL    12200
22.23.05.END     SCAT
22.23.05.REWIND(TAPE1)
22.23.05.RETURN(TAPE1)
22.23.05. TAPE LIMIT=0
22.23.05.REWIND(LGO)
22.23.05.RFL(75000)
22.23.05.CX    20.661 SEC.
22.23.05.PX   344.050 SEC.
22.23.05.NL    76000
22.23.08.REWIND(TAPE8)
22.23.08.LIBCOPY(SIATBIN,LGO,B=020)
22.23.11.LGO.
22.23.17.CX    22.142 SEC.
22.23.17.PX   349.914 SEC.
22.23.17.NL    57000
22.23.23.STOP
22.23.23.CP    25.054 SEC.
22.23.23.PP   351.285 SEC.
22.23.23.LINES =          0573 OCTAL
22.23.23.CM     2.561 MWD-SEC.
```


VARIABLE
2
VARIABLE 1
5.000
15.000
25.000
35.000
45.000
55.000
65.000
75.000
85.000
95.000
105.000
3.150
3.120
3.090
3.060
3.030
3.000
2.970
2.940
2.910
2.880
2.850
2.820
2.790
2.760
2.730
2.700
2.670
2.640
2.610
2.580
2.550
2.520
2.490
2.460
2.430
2.400
2.370
2.340
2.310
2.280
2.250
2.220
2.190
2.160
2.130
2.100
2.070
2.040
2.010
1.980
1.950
1.920
1.890
1.860
1.830
1.800
1.770
1.740
1.710
1.680
1.650

```
MM53880.   12/19/69.PURDUE MACE 11/27/69.

22.05.24.MM53A7     3512,NAM,S.160,CM76000,TP1.010
22.05.24..
22.05.24.MAP(ON)
22.05.24.RUN(S)
22.05.28.CTIME 000.213 SEC.  RUN MOD LEVEL 43
22.05.29.REQUEST(TAPE1,556,HY,X,C=RA,MT,READ)
22.16.09.  MT50 ASSIGNED - 556
22.16.10.REWIND(TAPE1)
22.16.10.LGO.
22.16.12.CX      .480 SEC.
22.16.12.PX     4.422 SEC.
22.16.12.NL    12200
22.23.05.END     SCAT
22.23.05.REWIND(TAPE1)
22.23.05.RETURN(TAPE1)
22.23.05. TAPE LIMIT=0
22.23.05.REWIND(LGO)
22.23.05.RFL(75000)
22.23.05.CX    20.661 SEC.
22.23.05.PX   344.050 SEC.
22.23.05.NL    76000
22.23.08.REWIND(TAPE8)
22.23.08.LIBCOPY(STAT9IN,LGO,AMD20)
22.23.11.LGO.
22.23.17.CX    22.142 SEC.
22.23.17.PX   349.914 SEC.
22.23.17.NL    57000
22.23.23.STOP
22.23.23.CP    25.054 SEC.
22.23.23.PP   351.285 SEC.
22.23.23.LINES =          0573 OCTAL
22.23.23.CM     2.501 MWD-SEC.
```

BMD PROGRAM TO PRODUCE SCATTERGRAM

EDITSYM CONTROL CARDS

```
*COPY,BMD2D,BMD5D

      PROGRAM BMD2D (INPUT=1,OUTPUT=1,TAPE4,TAPE5=INPUT,TAPE6=OUTPUT,    BMD2D00001
     1TAPE8)                                                             BMD2D00002
C     BMD02D      CORRELATION WITH TRANSGENERATION      NOVEMBER  13, 1964  BMD2D00003
C                                 0400 CONVERSION BETTY BENSON            BMD2D00004
C                          HEALTH SCIENCES COMPUTING FACILITY, UCLA        BMD2D00005
000003       TYPE INTEGER WK,ASK,A123,H123,C123,D123,A1,TJOB,CODE,REL,OP,FMT,BU  BMD2D00006
     1S                                                                  BMD2D00007
000003       TYPE INTEGER PC7,P01,P02,PW3                                BMD2D00008
000003       DIMENSION AMAX(150),AMIN(150),YY(15),SYM(120),NS(120),SX(150)  BMD2D00009
     1 ,SX2(150),SXY(135,135),DATA(150),FMT(120)                         BMD2D00010
000003       DIMENSION NEW(150),JUMP(150),NA(150),AN(150)                 BMD2D00011
000003       DIMENSION CON(36),NSUB(36),REL(36),OP(36),INDEX(36),WK(8)    BMD2D00012
000003       DIMENSION IDATA(14)                                         BMD2D00013
000003       COMMON SXY                                                  BMD2D00014
000003       COMMON IDATA                                                BMD2D00015
000003       DIMENSION C(255),Q(150)                                     BMD2D00016
C                                                                        BMD2D00017
000003   100 FORMAT(54H1BMD02D CORRELATION WITH TRANSGENERATION - VERSION OF  BMD2D00018
     A15HNOV. 13, 1964  /                                                BMD2D00019
     140H HEALTH SCIENCES COMPUTING FACILITY,UCLA//                      BMD2D00020
     214H PROBLEM CODE A6,/                                              BMD2D00021
     321H NUMBER OF VARIABLES I3//                                       BMD2D00022
     417H NUMBER OF CASES I5,//)                                         BMD2D00023
C                                                                        BMD2D00024
000003       PC7=(6HNO     )                                             BMD2D00025
000005       WK(1)=(6HGT     )                                           BMD2D00026
000006       WK(2)=(6HGE     )                                           BMD2D00027
000010       WK(3)=(6HLT     )                                           BMD2D00028
000011       WK(4)=(6HLE     )                                           BMD2D00029
000013       WK(5)=(6HEW     )                                           BMD2D00030
000014       WK(6)=(6HNE     )                                           BMD2D00031
000016       WK(7)=(6HOR     )                                           BMD2D00032
000017       WK(8)=(6HAN     )                                           BMD2D00033
000021       ASK=(6H**     )                                             BMD2D00034
000022       C123=(6HTRNGEN)                                             BMD2D00035
000024       D123=(6HPLOTSL)                                             BMD2D00036
000025       H123=(6HPROBLM)                                             BMD2D00037
000027       A123=(6HFINISH)                                             BMD2D00038
000030       A1=(6HV      )                                              BMD2D00039
000032       NTAPE=5                                                     BMD2D00040
000033       IT1=4                                                       BMD2D00041
C            IT1=1                                  (ORIGINAL)           BMD2D00042
000034   998 READ(            5,102)TJOB,CODE,NVAR,NSAM,NSEL,NADD,NADDL,PU1,PW2,  BMD2D00043
     1PW3,NTG,NTAPE,KVR                                                  BMD2D00044
000072       GO TO 996                                                   BMD2D00045
000073   999 DECODE(80,102,IDATA)TJOB,CODE,NVAR,NSAM,NSEL,NADD,NADDL,PU1,PW2,PW  BMD2D00046
     13,NTG,NTAPE,KVR                                                    BMD2D00047
000133   996 REWIND IT1                                                  BMD2D00048
000135       IF(IJOB-A123)700,701,700                                    BMD2D00049
000137   700 IF(IJOB-H123)9010,703,9010                                  BMD2D00050
000141   703 IF (NTAPE.EQ.0) 741,742                                     BMD2D00051
000145   741 NTAPE=5  $  GO TO 773                                       BMD2D00052
000147   742 IF (NTAPE.NE.4) GO TO 773                                   BMD2D00053
000151       PRINT 5                                                     BMD2D00054
000155     5 FORMAT (////////* YOU HAVE ASSIGNED LOGICAL NUMBER 4 TO YOUR DATA  BMD2D00055
     1 TAPE.  CHOOSE SOME OTHER NUMBER.*/////)                           BMD2D00056
```

```
000155      701 STOP                                                          RMD2D00057
000157      773 CONTINUE                                                      RMD2D00058
000157          IF((NVAR-1)*(136-NVAR)) 9000,9000,1008                        RMD2D00059
000164     1008 IF(NSAM-1)9001,9001,1002                                      RMD2D00060
000167     1002 IF((NVAR+NAUD-1)*(136-NVAR-NAUD)) 9002,9002,705              RMD2D00061
000176      705 IF(IABS (NBOOL)-9)706,706,9003                                RMD2D00062
000201      706 CALL VECHCK(KVR)                                              RMD2D00063
000203        3 WRITE(            6,100)CODE,NVAR,NSAM                        RMD2D00064
000215          IF (NTG-150) 1003,1003,9004                                   RMD2D00065
000220     1003 IF(NTG)9005,401,402                                           RMD2D00066
000222      402 WRITE(            6,403)                                      RMD2D00067
000226          WRITE(            6,404)                                      RMD2D00068
000232          DO 707 I=1,NTG                                              RMD2D00069
000234          READ(           5,406)TODE,NEW(I),JUMP(I),NA(I),BN(I)       RMD2D00070
000251          IF(TODE-C123)9006,405,9006                                  RMD2D00071
000253      405 WRITE(            6,4471I,NEW(I),JUMP(I),NA(I),BN(I)        RMD2D00072
000271          IF(JUMP(I)-41)2000,707,2005                                 RMD2D00073
000274     2000 IF(JUMP(I)*(17-JUMP(I)))2005,2005,707                       RMD2D00074
000301     2005 WRITE(            6,4001)                                     RMD2D00075
000305          JUMP(I)=99                                                  RMD2D00076
000307      707 CONTINUE                                                    RMD2D00077
000312      401 IF(NBOOL) 411,412,411                                       RMD2D00078
000313      411 KK=IABS (NBOOL)*4                                           RMD2D00079
000316          WRITE(            6,413)                                    RMD2D00080
000321          READ(           5,414)(NSUB(I),NEL(I),CON(I),OP(I),I=1,KK)  RMD2D00081
000342          WRITE(            6,415)                                    RMD2D00082
000346          DO 416 I=1,KK                                               RMD2D00083
000350          KK1=I                                                       RMD2D00084
000351          WRITE(            6,417)NSUB(I),NEL(I),CON(I),OP(I)         RMD2D00085
000365          IF(ASK-OP(I))416,1234,416                                   RMD2D00086
000370      416 CONTINUE                                                    RMD2D00087
000373     1234 DO 438 I=1,KK1                                              RMD2D00088
000375      438 INDEX(I)=0                                                  RMD2D00089
000401          DO 437 I=1,KK1                                              RMD2D00090
000402          DECODE(1,709,CON(I))PUS                                     RMD2D00091
000411      709 FORMAT(A1)                                                  RMD2D00092
000411          IF(A1-PUS)710,711,710                                       RMD2D00093
000413      711 DECODE(5,712,CON(I))CON(I)                                  RMD2D00094
000423      712 FORMAT(2X,F3.0)                                             RMD2D00095
000423          INDEX(I)=1                                                  RMD2D00096
000425          GO TO 437                                                   RMD2D00097
000426      710 DECODE(6,713,CON(I ))CON(I)                                 RMD2D00098
000436      713 FORMAT(F6.0)                                                RMD2D00099
000436      437 CONTINUE                                                    RMD2D00100
000441      412 IF(NTG*NBOOL) 423,418,425                                   RMD2D00101
000445      418 IF(NTG)9007,419,424                                         RMD2D00102
000447      419 IF(NBOOL)422,421,422                                        RMD2D00103
000450      421 JESUS=1                                                     RMD2D00104
000451          GO TO 7                                                     RMD2D00105
000452      422 JESUS=2                                                     RMD2D00106
000453          NOB=0                                                       RMD2D00107
000454          GO TO 7                                                     RMD2D00108
000455      424 JESUS=3                                                     RMD2D00109
000456          GO TO 7                                                     RMD2D00110
000457      425 JESUS=4                                                     RMD2D00111
000460          NOB=0                                                       RMD2D00112
000461          GO TO 7                                                     RMD2D00113
000462      423 JESUS=5                                                     RMD2D00114
000463          NOB=0                                                       RMD2D00115
000464        7 M=0                                                         RMD2D00116
000465          LCASE=0                                                     RMD2D00117
000466          LEFT=NSAM                                                   RMD2D00118
000467          NVAR1=NVAR+NAUD                                             RMD2D00119
000471          DO 4 I=1,NVAR1                                              RMD2D00120
```

```
000473          AMIN(I)=10.**10                                                  BMD2D00121
000476          AMAX(I)=-AMIN(I)                                                 BMD2D00122
000477          SX(I)=0.0                                                        BMD2D00123
000500          SX2(I)=0.0                                                       BMD2D00124
000501          DO 4 J=1,NVAR1                                                   BMD2D00125
000503        4 SXY(I,J)=0.0                                                     BMD2D00126
000513          KL=0                                                             BMD2D00127
000514          H=0.                                                             BMD2D00128
000515        6 KVR=KVR*8                                                        BMD2D00129
000517          READ(           5,103)(FMT(I),I=1,KVR)                           BMD2D00130
000531       77 DO 600 II=1,NSAM                                                 BMD2D00131
000533          READ(           MTAPE,FMT)(DATA(I),I=1,NVAR)                     BMD2D00132
000546          GO TO (407,427,428,429,430),JESUS                                BMD2D00133
000557      427 CALL COOL(NSUB,REL,CON,OP,INDEX,DATA,NTEST,WK,KK1,A123,B123,D123,N BMD2D00134
                1TAPE)                                                         BMD2D00135
000574          GO TO (600,431,999,701),NTEST                                    BMD2D00136
000604      431 NOB=NOB+1                                                        BMD2D00137
000606          GO TO 407                                                        BMD2D00138
000606      428 CALL TRNGEN(DATA,NVAR,NTG,NSAM,LEFT,LCASE,NF#,JUMP,NA,BN,M,II)   BMD2D00139
000622          IF(LCASE) 409,407,407                                            BMD2D00140
000624      409 LCASE=0                                                          BMD2D00141
000625          GO TO 600                                                        BMD2D00142
000626      429 CALL TRNGEN(DATA,NVAR,NTG,NSAM,LEFT,LCASE,NF#,JUMP,NA,BN,M,II)   BMD2D00143
000642          IF(LCASE) 409,433,433                                            BMD2D00144
000644      433 CALL COOL(NSUB,REL,CON,OP,INDEX,DATA,NTEST,WK,KK1,A123,B123,D123,N BMD2D00145
                1TAPE)                                                         BMD2D00146
000661          GO TO (600,431,999,701),NTEST                                    BMD2D00147
000671      430 CALL COOL(NSUB,REL,CON,OP,INDEX,DATA,NTEST,WK,KK1,A123,B123,D123,N BMD2D00148
                1TAPE)                                                         BMD2D00149
000706          GO TO (600,435,999,701),NTEST                                    BMD2D00150
000716      435 NOB=NOB+1                                                        BMD2D00151
000720          CALL TRNGEN(DATA,NVAR,NTG,NSAM,LEFT,LCASE,NF#,JUMP,NA,BN,M,II)   BMD2D00152
000733          IF(LCASE) 409,407,407                                            BMD2D00153
000735      407 H=H+1.                                                           BMD2D00154
000737          IF(H-1)13,101,13                                                 BMD2D00155
000741       101 HH=0.0                                                          BMD2D00156
000742          GO TO 112                                                        BMD2D00157
000743        13 HH=H/(H-1.)                                                     BMD2D00158
000746       112 DO 8 I=1,NVAR1                                                  BMD2D00159
000750          KL=KL+1                                                          BMD2D00160
000752          IF(KL-255)1004,1004,1005                                         BMD2D00161
000754      1005 WRITE(       IT1)C                                              BMD2D00162
000761          KL=1                                                             BMD2D00163
000762      1004 C(KL)=DATA(I)                                                   BMD2D00164
000765          AMIN(I)=AMIN1(AMIN(I),DATA(I))                                   BMD2D00165
000771          AMAX(I)=AMAX1(AMAX(I),DATA(I))                                   BMD2D00166
000775          SX(I)=SX(I)+DATA(I)                                              BMD2D00167
000777          Q(I)=DATA(I)-SX(I)/H                                             BMD2D00168
001003          QQ=Q(I)*HH                                                       BMD2D00169
001005          DO 8 J=1,I                                                       BMD2D00170
001006        8 SXY(I,J)=SXY(I,J)+Q(J)*QQ                                        BMD2D00171
001021       600 CONTINUE                                                        BMD2D00172
            ***** WARNING -- ORIGINAL PROGRAM HAD DIVIDE CHECK, ACCUMULATOR OVERFLOW BMD2D00173
            *****   QUOTIENT OVERFLOW TEST AT THIS POINT.  COMPILER HAS REPLACED IT W BMD2D00174
            *****   JUMP TO SECOND STATEMENT NUMBER, THUS IGNORING THE TEST.         BMD2D00175
001024          GO TO 1011                                                       BMD2D00176
001024      1011 DO 1012 I=1,NVAR1                                               BMD2D00177
001026          SX2(I)=SXY(I,I)                                                  BMD2D00178
001032          DO 1012 J=1,I                                                    BMD2D00179
001033      1012 SXY(J,I)=SXY(I,J)                                               BMD2D00180
001047          WRITE(       IT1)C                                               BMD2D00181
001054          GO TO (508,505,506,507,507),JESUS                                BMD2D00182
001065       505 NSAM=NOB                                                        BMD2D00183
```

```
001067          GO TO 505                                                BMD02D00184
001067     505  NSAM=LEFT                                                BMD02D00185
001071          GO TO 505                                                BMD02D00186
001071     507  NSAM=NSAM-(NSAM-LEFT)                                    BMD02D00187
001074     506  NVAR=NVAR1                                               BMD02D00188
001076          WRITE(          6,502)NSAM                               BMD02D00189
001103          IF(NSAM)500,500,501                                      BMD02D00190
001105     500  WRITE(          6,503)                                   BMD02D00191
001111     610  IF(NSEL)998,998,615                                      BMD02D00192
001113     615  DO 620 I=1,NSEL                                          BMD02D00193
001115     620  READ(          5, 1120)TOVE                              BMD02D00194
001125          GO TO 998                                                BMD02D00195
001125     501  END FILE IT1                                             BMD02D00196
001127          REWIND IT1                                               BMD02D00197
001131          WRITE(          6,104)                                   BMD02D00198
001135          WRITE(          6,105)(SA(I),I=1,NVAR)                   BMD02D00199
001150          WRITE(          6,108)                                   BMD02D00200
0011..          FN=NSAM                                                  BMD02D00201
001156          DO 10 I=1,NVAR                                           BMD02D00202
001157      10  DATA(I)=SX(I)/FN                                         BMD02D00203
001163          WRITE(          6,105)(DATA(I),I=1,NVAR)                 BMD02D00204
001176          IF(PU1-PC7) 50,51,50                                     BMD02D00205
001200      50  WRITE(          6,106)                                   BMD02D00206
001204          CALL PATTY(SXY,NVAR)                                     BMD02D00207
001206      51  IF(1.0-FN) 305,350,350                                   BMD02D00208
001211     305  WRITE(          6,109)                                   BMD02D00209
001215          DO 11 I=1,NVAR                                           BMD02D00210
001217          DATA(I)=SX2(I)/(FN-1.0)                                  BMD02D00211
001222      11  DATA(I)=SQRT (DATA(I))                                   BMD02D00212
001231          WRITE(          6,105)(DATA(I),I=1,NVAR)                 BMD02D00213
001243          DO 12 I=1,NVAR                                           BMD02D00214
001245          DO 12 J=1,NVAR                                           BMD02D00215
001246      12  SXY(I,J)=SXY(I,J)/(FN-1.0)                               BMD02D00216
001260          IF(PU2-PC7) 52,53,52                                     BMD02D00217
001262      52  WRITE(          6,110)                                   BMD02D00218
001266          CALL PATTY(SXY,NVAR)                                     BMD02D00219
001270      53  WRITE(          6,111)                                   BMD02D00220
001274          DO 14 I=1,NVAR                                           BMD02D00221
001276          DO 14 J=1,NVAR                                           BMD02D00222
001277      14  SXY(I,J)=SXY(I,J)/(DATA(I)*DATA(J))                      BMD02D00223
001313          CALL PATTY(SXY,NVAR)                                     BMD02D00224
001315     325  NPAGE=0                                                  BMD02D00225
001316          IF(NSEL)9000,998,201                                     BMD02D00226
```

```
001320     201  CALL PLOT(NVAR,NSAM,NSEL,AMAX,AMIN)                      BMD02D00227
001324          GO TO 998                                                BMD02D00228
          C                                                              BMD02D00229
001325     350  WRITE(               6,400)                              BMD02D00230
001331          GO TO 325                                                BMD02D00231
001332    9000  NI=773                                                   BMD02D00232
001333          NJ=1                                                     BMD02D00233
001334          GO TO 9011                                               BMD02D00234
001335    9001  NI=1006                                                  BMD02D00235
001336          NJ=0                                                     BMD02D00236
001337          GO TO 9011                                               BMD02D00237
001340    9002  NI=1002                                                  BMD02D00238
001341          NJ=0                                                     BMD02D00239
001342          GO TO 9011                                               BMD02D00240
001343    9003  NI=705                                                   BMD02D00241
001344          NJ=0                                                     BMD02D00242
001345          GO TO 9011                                               BMD02D00243
001346    9004  NI=3                                                     BMD02D00244
001347          NJ=1                                                     BMD02D00245
001350          GO TO 9011                                               BMD02D00246
001351    9005  NI=1003                                                  BMD02D00247
001352          NJ=0                                                     BMD02D00248
001353          GO TO 9011                                               BMD02D00249
001354    9006  NI=402                                                   BMD02D00250
001355          NJ=4                                                     BMD02D00251
001356          GO TO 9011                                               BMD02D00252
001357    9007  NI=418                                                   BMD02D00253
001360          NJ=0                                                     BMD02D00254
001361          GO TO 9011                                               BMD02D00255
001362    9008  NI=325                                                   BMD02D00256
001363          NJ=1                                                     BMD02D00257
001364          GO TO 9011                                               BMD02D00258
001365    9009  NI=703                                                   BMD02D00259
001366          NJ=0                                                     BMD02D00260
001367          GO TO 9011                                               BMD02D00261
001370    9010  NI=700                                                   BMD02D00262
001371          NJ=0                                                     BMD02D00263
001372    9011  WRITE(6,704)NI,NJ                                        BMD02D00264
001402          WRITE(6,901)                                             BMD02D00265
001406    9014  FORMAT(2U(120H                                           BMD02D00266
               1                                                         BMD02D00267
               2  ,/))                                                   BMD02D00268
001406          CALL EXIT                                                BMD02D00269
          C                                                              BMD02D00270
```

```
001407     102  FORMAT(2A6,I3,I5,I2,I4,I2,3A2    ,31X,I3,2I2)            BMD02D00271
001407     103  FORMAT(6A10)                                             BMD02D00272
001407     104  FORMAT(5H0SUMS//)                                        BMD02D00273
001407     105  FORMAT(1H 8F14.4)                                        BMD02D00274
001407     106  FORMAT(25H0CROSS PRODUCT DEVIATIONS)                     BMD02D00275
001407     108  FORMAT(6H0MEANS//)                                       BMD02D00276
001407     109  FORMAT(20H0STANDARD DEVIATIONS//)                        BMD02D00277
001407     110  FORMAT(27H0VARIANCE-COVARIANCE MATRIX//)                 BMD02D00278
001407     111  FORMAT(19H0CORRELATION MATRIX//)                         BMD02D00279
001407    1120  FORMAT(A6,I3,I2,20I3)                                    BMD02D00280
001407     403  FORMAT(1H06X,23HTRANS GENERATOR CARD(S))                 BMD02D00281
001407     404  FORMAT(46H0CARD     =EN      TRANS    ORIG.   ORIG. VAR(B)/45H  NO.   BMD02D00282
              1VARIABLE   CODE      VAR(S)   OR CONSTANT)                BMD02D00283
001407     406  FORMAT(A6,I3,I2,I3,F6.0)                                 BMD02D00284
001407     413  FORMAT(//21H CASE SELECTION CARDS//)                     BMD02D00285
001407     414  FORMAT(4(3X,I3,1X,A2,A6,1X,A2))                          BMD02D00286
001407     415  FORMAT(22H A CASE IS ACCEPTED IF)                        BMD02D00287
001407     417  FORMAT(6H (VAR(,I3,2H) ,A2,1X,A6,2H) ,A2)                BMD02D00288
```

```
001407      439 FORMAT(A6)                                                                  BMD2D00289
001407      442 FORMAT(F6.0)                                                                BMD2D00290
001407      443 FORMAT(2X,F3.0)                                                             BMD2D00291
001407      447 FORMAT(2H  12,I8,2I4,F14.4)                                                 BMD2D00292
001407      502 FORMAT(23HOREMAINING SAMPLE SIZE=I5)                                        BMD2D00293
001407      503 FORMAT(18HOND CASES ACCEPTED)                                               BMD2D00294
001407      704 FORMAT(46H CONTROL CARDS INCORRECTLY ORDERED OR PUNCHED.//33H ERRO          BMD2D00295
               1R FOUND AT STATEMENT NUMBER ,I4,2A3,5HPLUS ,I4,//26H IN MAIN PROGRA        BMD2D00296
               2M OF BMD02D)                                                               BMD2D00297
001407     4000 FORMAT(115HOTHE STANDARD DEVIATIONS, VARIANCE-COVARIANCE MATRIX AN         BMD2D00298
               XD CORRELATION MATRIX ARE UNDEFINED FOR A SAMPLE SIZE OF ONE.)              BMD2D00299
001407     4001 FORMAT(97HOINCORRECT TRANSGENERATION CODE ON CARD ABOVE. PROGRAM W         BMD2D00300
               XILL PROCEED WITHOUT THIS TRANSGENERATION.)                                 BMD2D00301
001407     1200 FORMAT(13A6,A2)                                                            BMD2D00302
           C                                                                               BMD2D00303
001407           END                                                                       BMD2D00304

PROGRAM LENGTH INCLUDING I/O BUFFERS
011070

BMD2D

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
3     - 000203    4     - 000503    5     - 001530    6     - 000815
7     - 000664    8     - 001006    10    - 001160    12    - 001246
13    - 000743    14    - 001277    50    - 001200    51    - 001206
52    - 001262    53    - 001270    77    - 000531    100   - 001423
101   - 000741    102   - 001576    103   - 001603    104   - 001605
105   - 001610    106   - 001613    108   - 001620    109   - 001623
110   - 001627    111   - 001634    112   - 000746    201   - 001320
305   - 001211    325   - 001315    350   - 001325    401   - 000312
402   - 000222    403   - 001643    404   - 001650    405   - 000253
406   - 001664    407   - 000735    409   - 000624    411   - 000313
412   - 000441    413   - 001667    414   - 001674    415   - 001700
416   - 000370    417   - 001704    418   - 000445    419   - 000647
421   - 000450    422   - 000452    423   - 000462    424   - 000455
425   - 000657    427   - 000557    428   - 000606    429   - 000626
430   - 000671    431   - 000604    433   - 000644    435   - 000718
437   - 000436    438   - 000376    439   - 001711    442   - 001713
443   - 001715    447   - 001720    500   - 001105    501   - 001125
502   - 001724    503   - 001731    505   - 001065    506   - 001067
507   - 001071    508   - 001074    600   - 001021    610   - 001111
615   - 001113    700   - 000137    701   - 000155    703   - 000141
704   - 001735    705   - 000176    706   - 000201    707   - 000307
709   - 001545    710   - 000426    711   - 000413    712   - 001547
713   - 001552    741   - 000145    742   - 000147    773   - 000157
996   - 000133    998   - 000034    999   - 000073    1000  - 002006
1002  - 000157    1003  - 000220    1004  - 000762    1005  - 000754
1008  - 000194    1011  - 001024    1012  - 001033    1120  - 001640
1234  - 000373    2000  - 000274    2005  - 000301    4000  - 001754
4001  - 001772    9000  - 001332    9001  - 001335    9002  - 001340
9003  - 001343    9004  - 001346    9005  - 001351    9006  - 001354
9007  - 001357    9008  - 001362    9009  - 001365    9010  - 001370
9011  - 001372    9014  - 001557

BLOCK NAMES AND LENGTHS
      - 043477
```

```
VARIABLE ASSIGNMENTS
AMAX   =  002412    AMIN   =  002640    ASK    =  002075    A1     =  002102
A123   =  002076    AN     =  005271    B123   =  002077    C      =  005673
CODE   =  002104    CON    =  005617    C123   =  002100    DATA   =  004141
D123   =  002101    FMT    =  002215    FN     =  006556    H      =  006545
HH     =  006550    I      =  006532    IDATA  =  043461C01  IT     =  006546
INDEX  =  005627    IT1    =  006521    J      =  006543    JESUS  =  006535
JUMP   =  004615    KK     =  006533    KK1    =  006534    KL     =  006544
KVH    =  006531    LCASE  =  006540    LEFT   =  006541    M      =  006537
MTAPE  =  006530    NA     =  005043    NADD   =  006525    NBODL  =  006526
NEW    =  004367    NI     =  006554    NJ     =  006555    NOG    =  006536
NPAGE  =  006553    NS     =  003275    NSAM   =  006523    NSEL   =  006524
NSUB   =  005563    NTAPE  =  006520    NTEST  =  006547    NTR    =  006527
NVAR   =  006542    NVAR1  =  006542    OP     =  002151    PQ2    =  002406
PQ1    =  002407    PW2    =  002410    PW3    =  002411    PW6    =  002405
Q      =  006272    QQ     =  006551    REL    =  002105    SX     =  003465
SXY    =  000000C01 SX2    =  003713    SYM    =  003105    TONE   =  002103
WK     =  002065    YY     =  003066

START OF CONSTANTS
001411

START OF TEMPORARIES
002011

START OF INDIRECTS
002051

UNUSED COMPILER SPACE
014700
```

```
                SUBROUTINE COOL (NSUB,REL,CON,OP,INDEX,DATA,NTEST,WK,KK1,A123,D123,    RMD2D00305
               1D123,NTAPE)                                                           RMD2D00306
      CCOOL           SUBROUTINE COOL FOR RMD02D                                      RMD2D00307
      C               REWRITTEN BY DU BOIS      3-25-64                               RMD2D00308
      C                                                                               RMD2D00309
      C     DEFINITION OF NTEST:                                                      RMD2D00310
      C     NTEST=1 IF CASE FAILS BOOLEAN TEST                                        RMD2D00311
      C     NTEST=2 IF CASE SATISFIES BOOLEAN TEST                                    RMD2D00312
      C     NTEST=3 IF NEW PROBLEM CARD IS DETECTED                                   RMD2D00313
      C     NTEST=4 IF FINISH CARD IS DETECTED                                        RMD2D00314
      C                                                                               RMD2D00315
000020        TYPE INTEGER WK,OP,REL,A123,B123,D123,X                                 RMD2D00316
000020        DIMENSION NSUB(36),REL(36),CON(36),OP(36),INDEX(36),WK( 8),IDATA(15     RMD2D00317
             1O), SXY(135,135),IN(37),IDATA(14)                                       RMD2D00318
      C                                                                               RMD2D00319
000020        COMMON SXY,IDATA                                                        RMD2D00320
000020        DO 100 I=1,KK1                                                          RMD2D00321
000021        IS=NSUB(I)                                                              RMD2D00322
      C                                                                               RMD2D00323
      C     EXAMINE BOOLEAN EXPRESSION FOR GT, GE, LT, LE, EQ, NE                     RMD2D00324
      C                                                                               RMD2D00325
000023        DO 55 J=1,6                                                             RMD2D00326
000025        IF(REL(I)-WK(J))55,26,55                                                RMD2D00327
000031     55 CONTINUE                                                                RMD2D00328
000033        GO TO 311                                                               RMD2D00329
000034     26 IF(INDEX(I)) 27,27,28                                                   RMD2D00330
000037     27 CC=CON(I)                                                               RMD2D00331
000042        GO TO 29                                                                RMD2D00332
000042     28 K=CON(I)                                                                RMD2D00333
000045        CC=DATA(K)                                                              RMD2D00334
000047     29 K=DATA(IS)-CC                                                           RMD2D00335
```

```
000052          IF((CC.EQ.0.).OR.(DATA(I5).EQ.0.))299,2300                      RMD2D00336
000061     2300 A=ABS (B/CC)-2**(-35)                                           RMD2D00337
000071          GO TO(1,2,3,4,5,6),J                                            RMD2D00338
000103     2299 GO TO(11,12,13,14,15,16),J                                      RMD2D00339
000115        1 IF(A)50,50,11                                                   RMD2D00340
000117       11 IF(B)50,50,20                                                   RMD2D00341
000121        2 IF(A)20,20,12                                                   RMD2D00342
000123       12 IF(B)50,20,20                                                   RMD2D00343
000125        3 IF(A)50,50,13                                                   RMD2D00344
000127       13 IF(B)20,50,50                                                   RMD2D00345
000131        4 IF(A)20,20,14                                                   RMD2D00346
000133       14 IF(B)20,20,50                                                   RMD2D00347
000135        5 IF(A)20,20,15                                                   RMD2D00348
000137       15 IF(B)50,20,50                                                   RMD2D00349
000141        6 IF(A)50,50,15                                                   RMD2D00350
000143       16 IF(B)20,50,20                                                   RMD2D00351
000144       20 IN(I)=1                                                         RMD2D00352
000147          GO TO 100                                                       RMD2D00353
000147       50 IN(I)=0                                                         RMD2D00354
000151      100 CONTINUE                                                        RMD2D00355
000154          NTEST=IN(1)                                                     RMD2D00356
000155          KK=KK1-1                                                        RMD2D00357
000157          IF(KK)500,500,501                                               RMD2D00358
                C                                                             RMD2D00359
                C       EXAMINE BOOLEAN OPERATOR FOR OR/AN                    RMD2D00360
                C                                                             RMD2D00361
000160       501  DO 200 I=1,KK                                               RMD2D00362
000162            IF(OP(I)-WK(7))222,191,222                                    RMD2D00363
000166       191  IF(NTEST) 199,199,321                                         RMD2D00364
000170       199  NTEST=IN(I+1)                                                 RMD2D00365
000173            GO TO 200                                                     RMD2D00366
000173       222  IF(OP(I)-WK(8))301,223,301                                    RMD2D00367
000177       223  NTEST=NTEST*IN(I+1)                                           RMD2D00368
000203       200  CONTINUE                                                      RMD2D00369
000206       500  IF(NTEST)320,320,321                                          RMD2D00370
000210       321  NTEST=2                                                       RMD2D00371
000212            GO TO 333                                                     RMD2D00372
000212       320  NTEST=1                                                       RMD2D00373
000214            GO TO 333                                                     RMD2D00374
                C                                                             RMD2D00375
                C       ERROR LOOK FOR NEXT PROBLEM OR FINISH CARD            RMD2D00376
                C       A123=6HFINISH                                         RMD2D00377
                C       B123=6HPROBLM                                         RMD2D00378
                C       D123=6HPLOTSL                                         RMD2D00379
                C                                                             RMD2D00380
000214       341  X=REL(I)                                                      RMD2D00381
000217            GO TO 313                                                     RMD2D00382
000217       301  X=OP(I)                                                       RMD2D00383
000222       313  WRITE(              6,2000)X                                  RMD2D00384
000230            IF(NTAPE-5) 302,302,304                                       RMD2D00385
000237       302  J=NTAPE                                                       RMD2D00386
000241       312  READ(J,1000)(IDATA(K),K=1,14)                                 RMD2D00387
000253            IF(IDATA-D123)305,312,305                                     RMD2D00388
000261       305  IF(IDATA-B123)307,306,307                                     RMD2D00389
000263       307  IF(IDATA-A123)312,309,312                                     RMD2D00390
000266       306  NTEST=3                                                       RMD2D00391
000270            GO TO 333                                                     RMD2D00392
000270       309  NTEST=4                                                       RMD2D00393
000272            GO TO 333                                                     RMD2D00394
000272       304  REWIND NTAPE                                                  RMD2D00395
000274            J=5                                                           RMD2D00396
000275            GO TO 312                                                     RMD2D00397
```

```
000276      333 RETURN                                                                  BMD20D0398
000277     1000 FORMAT(13A6,A2)                                                         BMD20D0399
000277     2000 FORMAT(31H1ILLEGAL OPERATOR OR RELATION ,A2,5H  IN CASE SELECTIO       BMD20D0400
               1N CARD, PROGRAM SKIPPED TO NEXT PROBLEM.)                            BMD20D0401
000277          END                                                                    BMD20D0402

SUBPROGRAM LENGTH
000416

COOL

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1       = 000115    2       = 000121    3       = 000125    4       = 000131
5       = 000135    6       = 000141    11      = 000117    12      = 000123
13      = 000127    14      = 000133    15      = 000137    16      = 000143
20      = 000144    26      = 000034    27      = 000037    28      = 000042
29      = 000047    50      = 000147    52      = 000031    100     = 000151
191     = 000166    199     = 000170    200     = 000203    222     = 000173
223     = 000177    301     = 000217    302     = 000237    304     = 000272
305     = 000261    306     = 000266    307     = 000263    308     = 000270
311     = 000214    312     = 000241    313     = 000222    320     = 000212
321     = 000210    333     = 000276    500     = 000206    501     = 000160
1000    = 000303    2000    = 000306    2099    = 000103    2300    = 000061

BLOCK NAMES AND LENGTHS
        = 043477

VARIABLE ASSIGNMENTS
A       = 000406    A123    = 000003    B       = 000405    B123    = 000004
CC      = 000603    C123    = 000005    I       = 000400    IDATA   = 043461C01
IN      = 000333    IS      = 000401    J       = 000402    K       = 000404
KK      = 000407    KK1     = 000002    NTAPE   = 000006    NTEST   = 000000
SAY     = 000000C01 WK      = 000001    X       = 000332

START OF CONSTANTS
000301

START OF TEMPORARIES
000322

START OF INDIRECTS
000330

UNUSED COMPILER SPACE
021300

                SUBROUTINE CONV(CC, SYM, KK)                                        BMD20D0403
          CCONV     SUBROUTINE CONV FOR BMD02D (3600 FORTRAN VERSION)               BMD20D0404
000006          DIMENSION CC(13005), SYM(65), CHAR(6)                               BMD20D0405
000006          TYPE INTEGER  CC, SYM, CHAR                                         BMD20D0406
000006          DO 2 I = 1, KK                                                      BMD20D0407
000007          DECODE (6,10,CC(I)) (CHAR(J),J=1,6)                                 BMD20D0408
000027       10 FORMAT (6R1)                                                        BMD20D0409
000027          DO 1 J = 1, 6                                                       BMD20D0410
000033          CHAR(J)=SYM(65-CHAR(J))                                             BMD20D0411
000036        1 CONTINUE                                                            BMD20D0412
000040        2 ENCODE(6,20,CC(I)) (CHAR(J),J=1,6)                                  BMD20D0413
000066       20 FORMAT (6A1)                                                        BMD20D0414
000066          RETURN                                                              BMD20D0415
000066          END                                                                 BMD20D0416

SUBPROGRAM LENGTH
000112
```

```
CONV

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10      =  000071    20      =  000073

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
CHAR    =  000102    I       =  000110    J       =  000111

START OF CONSTANTS
000070

START OF TEMPORARIES
000075

START OF INDIRECTS
000100

UNUSED COMPILER SPACE
022400
```



```
                SUBROUTINE PATTY(A,N)                                                   RMD2D00417
        CPATTY          SUBROUTINE PATTY FOR BMJ02D     RECOMPILATION DATE 8=26=63      RMD2D00418
        C                                                                               RMD2D00419
000005          DIMENSION A(135,135),NN(8)                                              RMD2D00420
        C                                                                               RMD2D00421
000005          IT=1                                                                    RMD2D00422
000006          KK=0                                                                    RMD2D00423
000007          K1=IT                                                                   RMD2D00424
000010          K2= MIN0 (8,N)                                                          RMD2D00425
000013        5 KK=KK+8                                                                 RMD2D00426
000015          IF(N-KK)3,3,4                                                           RMD2D00427
000017        4 IT=IT+1                                                                 RMD2D00428
000021          GO TO 5                                                                 RMD2D00429
000021        3 DO 50 JX=1,IT                                                           RMD2D00430
000023          LLL=K2-K1+1                                                             RMD2D00431
000026          LL=0                                                                    RMD2D00432
000027          DO 40 JJ=K1,K2                                                          RMD2D00433
000030          LL=LL+1                                                                 RMD2D00434
000032       40 NN(LL)=JJ                                                               RMD2D00435
000036          WRITE(          6,300)(NN(II),II=1,LLL)                                 RMD2D00436
000050          DO 10 I=1,N                                                             RMD2D00437
000053       10 WRITE(          6,20)I,(A(I,J),J=K1,K2)                                 RMD2D00438
000076          K1=K2+1                                                                 RMD2D00439
000100          K2=K1+7                                                                 RMD2D00440
000101          K2= MIN0 (K2,N)                                                         RMD2D00441
000104       20 FORMAT(1H I3,F11.4,7F14.4)                                              RMD2D00442
000104      300 FORMAT(1H0BX,4HCOL,7(10X,4HCOL),/,6X,I3,7( 11X,I3),/,4H ROW//)          RMD2D00443
000104       50 CONTINUE                                                               RMD2D00444
000107          RETURN                                                                  RMD2D00445
000107          END                                                                     RMD2D00446

SUBPROGRAM LENGTH
000157
```

```
PATTY

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
3       = 000041     4       = 000017     5       = 000013     20      = 000113
300     = 000117

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
I       = 000155     II      = 000154     IJ      = 000144     J       = 000156
JJ      = 000153     JX      = 000150     KK      = 000145     KJ      = 000146
K2      = 000147     LL      = 000152     LLL     = 000151     NV      = 000134

START OF CONSTANTS
000111

START OF TEMPORARIES
000126

START OF INDIRECTS
000133

UNUSED COMPILER SPACE
022300
```



```
                SUBROUTINE PLOT(NV,KD,NPL,K,AMIN)                                  BMD20D00447
          CPLOT       SUBROUTINE FOR BMD02D       NOV 3, 1964                      BMD20D00448
          C                                                                        BMD20D00449
000010            TYPE INTEGER A1,A2,A3,PL,A5,A6,A7,A8,A9,A10,A11,CM,CCC,C4,CJ,STM  BMD20D00450
                 1 ,Z,CC,JC,P,A4,HUF,DD,C                                          BMD20D00451
000010            TYPE INTEGER CM,CCC                                              BMD20D00452
000010            DIMENSION CIC(255),HUF(4)                                        BMD20D00453
000010            DIMENSION CC(13005),IC(13005),CJ(6),CM(6),IX(150),IY(150),X(150),C BMD20D00454
                 1(255),MM(20),AMIN(150),V(1500),T(11),DD(867,15),STM(45),JC(6)    BMD20D00455
          C                                                                        BMD20D00456
000010            COMMON CC,X,C,IX,IY,MM,T,STM,CJ,CM                               BMD20D00457
000010            EQUIVALENCE (CC,IC,DD),(CJ,JC)                                   BMD20D00458
000010            EQUIVALENCE(CIC,C)                                               BMD20D00459
000010            NV1=NV+1                                                         BMD20D00460
000012            A1=(+6H//$$$$)                                                   BMD20D00461
000013            A2=(+6H$$$$**)                                                   BMD20D00462
000015            A3=(+6H******)                                                   BMD20D00463
          CCCCCCCCCCCCCCCCCCCC  CHANGE FOR NU...MARCH 1968 CCCCCCCCCCCC            BMD20D00464
          C         A4=(+6H++++++)                                                 BMD20D00465
000016            A4=(+6H+ +++-)                                                   BMD20D00466
000020            PL=(+6HPLOTSL)                                                   BMD20D00467
000021            A5=6H-----Z                                                      BMD20D00468
000023            A6=6HYX=VUT                                                      BMD20D00469
000024            A7=6HSRQPON                                                      BMD20D00470
000026            A8=6HMLKJIH                                                      BMD20D00471
000027            A9=6HGFEDCH                                                      BMD20D00472
000031            A10=6HA98765                                                     BMD20D00473
000032            A11=6H4321                                                       BMD20D00474
          C  BATTELLE COMMENT  INCREASE OCTAL WORD SIZE TO 20 DIGITS               BMD20D00475
000034            CM=37000000000000000000B                                         BMD20D00476
000035            CCC=40000000000000000000B                                        BMD20D00477
000037            DO 800 I=1,NV                                                    BMD20D00478
000040            CALL SCALE(AMIN(I),R(I),100.,JJJJ,AMIN(I),R(I),HHH)              BMD20D00479
000052     800    R(I)=R(I)-AMIN(I)                                                BMD20D00480
          C  BATTELLE COMMENT  INCREASE OCTAL WORD SIZE TO 20 DIGITS               BMD20D00481
```

```
000062          CM(1)=77000000000000000000B                                     BMD2D00482
000063          CM(2)=00770000000000000000B                                     BMD2D00483
000065          CM(3)=00007700000000000000B                                     BMD2D00484
000066          CM(4)=00000077000000000000B                                     BMD2D00485
000070          CM(5)=00000000770000000000B                                     BMD2D00486
000071          CM(6)=00000000007700000000B                                     BMD2D00487
000073          C1=6HAAAAAA                                                     BMD2D00488
000074          C4=(+6H+++++)                                                   BMD2D00489
000076          DO 31 I=1,6                                                     BMD2D00490
000077       31 CJ(I)=C1.AND.CM(I)                                              BMD2D00491
000105          IT=4                                                            BMD2D00492
000106          ENCODE(66,400,BUF)A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11            BMD2D00493
000141          DECODE(65,401,BUF)(SYM(I),I=1,65)                               BMD2D00494
000160           K=(K-1)/6+1                                                    BMD2D00495
000164      400 FORMAT(11A6)                                                    BMD2D00496
000164      401 FORMAT(72A1)                                                    BMD2D00497
000164      96  RE#IND IT                                                      BMD2D00498
000166          K=0                                                             BMD2D00499
000167          DO 570 II=1,NPL                                                 BMD2D00500
000176          READ(5,5000) (X(I),I=1,8)                                       BMD2D00501
000205      5000 FORMAT (8A10)                                                  BMD2D00502
000205          DECODE(72,2,A)Z,LL,N,(MM(I),I=1,20)                             BMD2D00503
000232      2   FORMAT(A6,I3,I2,20I3)                                           BMD2D00504
000232          IF(LL*(NV1-LL))3,5,77                                           BMD2D00505
000241       77 IF(Z-PL)3,5,3                                                   BMD2D00506
000243        3 WRITE(          6,10)II,(X(I),I=1,12)                         BMD2D00507
000257      10  FORMAT(24HOERROR ON SELECTION CARD13,5X,12A6)                   BMD2D00508
000257          GO TO 570                                                       BMD2D00509
000263      5   DO 6 I=1,20                                                     BMD2D00510
000265          IF((NV-MM(I))*MM(I))3,7,7                                       BMD2D00511
000271      7   IF(MM(I))770,770,8                                              BMD2D00512
000274      8   K=K+1                                                           BMD2D00513
000276          IX(K)=MM(I)                                                     BMD2D00514
000301      6   IY(K)=LL                                                        BMD2D00515
000305       770 IF((NPL-II)*(K-60))570,600,600                                 BMD2D00516
000312      600 KN=K                                                            BMD2D00517
000314          L1=0                                                            BMD2D00518
000315      510 L0=L1+1                                                         BMD2D00519
000317          L1=MIN0 (L1+15,KN)                                              BMD2D00520
000323          K=0                                                             BMD2D00521
000324          DO 500 L=L0,L1                                                  BMD2D00522
000326          K=K+1                                                           BMD2D00523
000330          IX(K)=IX(L)                                                     BMD2D00524
000333      500 IY(K)=IY(L)                                                     BMD2D00525
000337      9   KK=867*K                                                        BMD2D00526
000342          DO 32 N=1,KK                                                    BMD2D00527
000343      32  IC(N)=0                                                         BMD2D00528
000347          KL=255                                                          BMD2D00529
000350          DO 13 JJ=1,N0                                                   BMD2D00530
000352      301 DO 12 J=1,NV                                                    BMD2D00531
000354          KL=KL+1                                                         BMD2D00532
000356          IF(KL-255)12,12,14                                              BMD2D00533
000360       14 READ(      IT)CIC                                             BMD2D00534
000365          KL=1                                                            BMD2D00535
000366       12 X(J)=CIC(KL)                                                    BMD2D00536
000377      302 DO 13 J=1,K                                                     BMD2D00537
000401          L=IX(J)                                                         BMD2D00538
000403          M=IY(J)                                                         BMD2D00539
000405          L=51.5-(X(L)-AMIN(L))/R(L)*50.                                  BMD2D00540
000415          IF (52-L)13,13,70                                               BMD2D00541
000417      70  M=(X(M)-AMIN(M))/R(M)*100.+1.5                                 BMD2D00542
000427          IF((102-M)*M)13,13,71                                           BMD2D00543
000432      71  M1=MOD (M-1,6)+1                                               BMD2D00544
000440          M2=(M+5)/6                                                      BMD2D00545
```

```
000444          M3=L+51*(M2+17*J-18)                                          BMD2D00546
000452          IF(M1-1) 60,60,61                                             BMD2D00547
000455       60 MUG=CM(M1).AND..NOT.CC(M3)                                    BMD2D00548
000460          IF(MUG)61,63,61                                               BMD2D00549
000461       63 IF(CC(M3))13,64,64                                            BMD2D00550
000464       64 CC(M3)=CC(M3).AND..NOT.CM.OR.CCC                              BMD2D00551
000471          GO TO 13                                                      BMD2D00552
000471       61 MUG=CM(M1).AND..NOT.CC(M3)                                    BMD2D00553
000476          IF(MUG)16,13,16                                               BMD2D00554
000477       16 IC(M3)=ISIGN (IABS (IC(M3))+JC(M1)*IC(M3))                    BMD2D00555
000506       13 CONTINUE                                                      BMD2D00556
000513      701 READ IT                                                       BMD2D00557
000515          CALL CONV(CC,SYM,KK)                                          BMD2D00558
000520          P=(+1H )                                                      BMD2D00559
000522          DO 40 N=1,101                                                 BMD2D00560
000526       40 C(N)=P                                                        BMD2D00561
000533          DO 41 N=1,101,5                                               BMD2D00562
000534       41 C(N)=C4                                                       BMD2D00563
000541          DO 50 J=1,K                                                   BMD2D00564
000542          L=IX(J)                                                       BMD2D00565
000544          M=IY(J)                                                       BMD2D00566
000546          Q=AMIN(M)                                                     BMD2D00567
000550          D=R(M)/10.                                                    BMD2D00568
000552          DO 51 N=1,11                                                  BMD2D00569
000553          T(N)=Q                                                        BMD2D00570
000555       51 Q=Q+D                                                         BMD2D00571
000560          Q=AMIN(L)+R(L)                                                BMD2D00572
000564          D=R(L)/50.                                                    BMD2D00573
000566          DO 52 N=1,51                                                  BMD2D00574
000567          X(N)=Q                                                        BMD2D00575
000571       52 Q=Q-D                                                         BMD2D00576
000574       50 WRITE(          6,54)*L,(T(N),N=1,11,2),(T(N),N=2,10,2),(C(N),N BMD2D00577
                1=1,101),(X(K00),C(K00),(DD(N,J),N=K00,867,51)),C(K00),      K00=1, BMD2D00578
                251),(C(N),N=1,101),(T(N),N=1,11,2),(T(N),N=2,10,2)           BMD2D00579
000702       54 FORMAT(11HI  VARIABLE*8X,8HVARIABLE13/17/2X,F15.3,5F20.3/7X,5F20.3 BMD2D00580
                1/13A,101A1,51(/1X,F10.3,1X,A1,16A6,A5,A1       )/13X,101A1/2X,F15.3 BMD2D00581
                2,5F20.3/7X,5F20.3)                                           BMD2D00582
000702          IF(L1-KN)510,580,580                                          BMD2D00583
000704      580 K=0                                                           BMD2D00584
000705      570 CONTINUE                                                      BMD2D00585
000710      900 RETURN                                                        BMD2D00586
000711          END                                                           BMD2D00587

SUBPROGRAM LENGTH
001137
PLOT

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
2     - 000770   3     - 000243   5     - 000263   7     - 000271
8     - 000274   9     - 000337   10    - 000774   12    - 000346
13    - 000506   14    - 000360   16    - 000477   32    - 000100
32    - 000344   40    - 000527   41    - 000535   54    - 001012
60    - 000455   61    - 000471   63    - 000461   64    - 000464
70    - 000417   71    - 000432   77    - 000241   96    - 000164
301   - 000352   302   - 000377   400   - 000761   401   - 000763
510   - 000315   570   - 000705   580   - 000704   600   - 000312
701   - 000513   770   - 000305   900   - 000710   5000  - 000766

BLOCK NAMES AND LENGTHS
      - 032772
```

```
VARIABLE ASSIGNMENTS
A1      =  001053    A10     =  001064    A11     =  001065    A2      =  001054
A3      =  001055    A4      =  001072    A5      =  001057    A6      =  001060
A7      =  001061    A8      =  001062    A9      =  001063    BUF     =  001073
C       =  031543C01 CC      =  000000C01 CCC     =  001105    CTR     =  031543C01
CJ      =  032756C01 CM      =  032764C01 CMM     =  001104    C1      =  001066
C4      =  001057    D       =  001135    DD      =  000000C01 HHH     =  001111
I       =  001107    IC      =  000000C01 II      =  001114    IT      =  001112
IX      =  032142C01 IY      =  032970C01 J       =  001126    JC      =  032756C01
JJ      =  001125    JJJJ    =  001110    K       =  001113    KK      =  001123
KL      =  001124    KN      =  001117    KVO     =  001136    L       =  001122
LL      =  001115    L0      =  001121    L1      =  001120    M       =  001127
MM      =  032C16C01 MUG     =  001133    M1      =  001130    M2      =  001131
M3      =  001132    N       =  001116    NV1     =  001106    P       =  001071
PL      =  001056    Q       =  001134    STM     =  032655C01 T       =  032642C01
X       =  031315C01 Z       =  001070

START OF CONSTANTS
000713

START OF TEMPORARIES
001032

START OF INDIRECTS
001046

UNUSED COMPILER SPACE
017500
```



```
                SUBROUTINE TRNGEN(DATA,IND,NVG,NUDATA,ISAMP,_CASE,NNEWA,LLCODE,    BMD20005RH
               1 LLVA,BBNEW,MERRY,N)                                               BMD20005RQ
          CTRNGEN        SUBROUTINE TRNGEN FOR BMD020          NOVEMBER 13, 1964  BMD20005QQ
          C                                                                        BMD20005Q1
          C                                                                        BMD20005Q2
          C                                                                        BMD20005Q3
000017            DIMENSION DATA(150),NNEWA(150),LLCODE(150),LLVA(150),BBNEW(150)  BMD20005Q4
000017            DIMENSION VECTOR(135,135)                                        BMD20005Q5
000017            COMMON VECTOR                                                    BMD20005Q6
          C                                                                        BMD20005Q7
000017            ASNF(XX)=ATAN (XX/SQRT (1.0-XX**2))                              BMD20005Q8
000036            ITEM=N                                                           BMD20005Q9
000040            SAMP=NUDATA                                                      BMD20006Q0
000041            DO 3 J=1,NVG                                                     BMD20006Q1
000043     305    NEWA=NNEWA(J)                                                    BMD20006Q2
000046            LCODE=LLCODE(J)                                                  BMD20006Q3
000051     310    LVA=LLVA(J)                                                      BMD20006Q4
000054            BNEW=BBNEW(J)                                                    BMD20006Q5
000057     315    IF(LCODE-10) 4,4,5                                               BMD20006Q6
000062          5 NEWB=BNEW                                                        BMD20006Q7
000064          4 D=DATA(LVA)                                                      BMD20006Q8
000067            IF(LCODE-41)500,170,3                                          BMD20006Q9
000071      500   GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,         BMD20006R0
                 1150,160),LCODE                                                 BMD20006R1
000115       10   IF(D)99,7,8                                                    BMD20006R2
000117        7   DATA(NEWA)=0.0                                                 BMD20006R3
000121            GO TO 3                                                        BMD20006R4
000122        8   DATA(NEWA)=SQRT (D)                                            BMD20006R5
000132            GO TO 3                                                        BMD20006R6
000133       20   IF(D)99,11,12                                                  BMD20006R7
000135       11   DATA(NEWA)=1.0                                                 BMD20006R8
000140            GO TO 3                                                        BMD20006R9
000140       12   DATA(NEWA)=SQRT (D)+SQRT (D+1.0)                               BMD20006S0
000161            GO TO 3                                                        BMD20006S1
```

```
000161      30 IF(D)99,99,14                                                    RMD2D00622
000163      14 DATA(NEWA)=ALOG(D)*.4342944819                                   RMD2D00623
000174         GO TO 3                                                          RMD2D00624
000174      40 DATA(NEWA)=EXP (D)                                               RMD2D00625
000204         GO TO 3                                                          RMD2D00626
000205      50 IF(D)99,7,17                                                     RMD2D00627
000207      17 IF(D-1.0)18,19,99                                                RMD2D00628
000212      19 DATA(NEWA)=3.14159265/2.0                                        RMD2D00629
000215         GO TO 3                                                          RMD2D00630
000215      18 A=SQRT (D)                                                       RMD2D00631
000220         DATA(NEWA)=ASNF(A)                                               RMD2D00632
000230         GO TO 3                                                          RMD2D00633
000230      60 A=D/(SAMP+1.0)                                                   RMD2D00634
000233         B=A+1.0/(SAMP+1.0)                                               RMD2D00635
000237         IF(A)99,23,24                                                    RMD2D00636
000240      23 IF(B)99,7,27                                                     RMD2D00637
000242      27 DATA(NEWA)=ASNF(SQRT (B))                                        RMD2D00638
000255         GO TO 3                                                          RMD2D00639
000255      24 IF(B)99,28,29                                                    RMD2D00640
000257      28 DATA(NEWA)=ASNF(SQRT (A))                                        RMD2D00641
000272         GO TO 3                                                          RMD2D00642
000272      29 A=SQRT (A)                                                       RMD2D00643
000275         B=SQRT (B)                                                       RMD2D00644
000277         DATA(NEWA)=ASNF(A)+ASNF(B)                                       RMD2D00645
000313         GO TO 3                                                          RMD2D00646
000313      70 IF(D)31,99,31                                                    RMD2D00647
000314      31 DATA(NEWA)=1.0/D                                                 RMD2D00648
000317         GO TO 3                                                          RMD2D00649
000320      80 DATA(NEWA)=D+BNEW                                                RMD2D00650
000324         GO TO 3                                                          RMD2D00651
000324      90 DATA(NEWA)=D*BNEW                                                RMD2D00652
000327         GO TO 3                                                          RMD2D00653
000330     100 IF(D)33,7,33                                                     RMD2D00654
000331      33 DATA(NEWA)=D**BNEW                                               RMD2D00655
000336         GO TO 3                                                          RMD2D00656
000337     110 DATA(NEWA)=D+DATA(NEWB)                                          RMD2D00657
000344         GO TO 3                                                          RMD2D00658
000344     120 DATA(NEWA)=D-DATA(NEWB)                                          RMD2D00659
000351         GO TO 3                                                          RMD2D00660
000351     130 DATA(NEWA)=D*DATA(NEWB)                                          RMD2D00661
000356         GO TO 3                                                          RMD2D00662
000356     140 IF(DATA(NEWB))34,99,34                                           RMD2D00663
000360      34 DATA(NEWA)=D/DATA(NEWB)                                          RMD2D00664
000365         GO TO 3                                                          RMD2D00665
000365     150 BNEW=NEWB                                                        RMD2D00666
000367         IF(D-BNEW)7,11,11                                                RMD2D00667
000372     160 IF(D-DATA(NEWB))7,11,11                                          RMD2D00668
000376     170 IF(D) 3,503,3                                                    RMD2D00669
000377     503 IF(SIGN (10.0,D)) 504,3,3                                        RMD2D00670
000403     504 DATA(NEWA)=BNEW                                                  RMD2D00671
000406       3 CONTINUE                                                         RMD2D00672
000411         GO TO 42                                                         RMD2D00673
000411      99 LCASE=-999                                                       RMD2D00674
000412         IF(MERRY-J) 402,401,402                                          RMD2D00675
000415     402 MERRY=J                                                          RMD2D00676
000417         WRITE(              6,1404)J                                     RMD2D00677
000425     401 WRITE(              6,1405)ITEM                                  RMD2D00678
000433         WRITE(              6,1406)                                      RMD2D00679
000437         ISAMP=ISAMP-1                                                    RMD2D00680
000444      42 RETURN                                                           RMD2D00681
000445    1404 FORMAT(30H0THE INSTRUCTIONS INDICATED ON/25H TRANS GENERATOR CARD RMD2D00682
           1NO.,I2,4H RE-/29H SULTED IN THE VIOLATION OF A/31H RESTRICTION FOR RMD2D00683
           2THIS TRANSFOR-/31H MATION. THE VIOLATION OCCURRED/27H FOR THE CASE RMD2D00684
           3 LISTED BELOW./)                                                  RMD2D00685
```

```
000445     1405 FORMAT( 9H CASE NO.,I5)                                          RMD2000686
000445     1408 FORMAT(45HOTHIS CASE WILL BE DELETED FOR ALL VARIABLES )         RMD2000687
          C                                                                      RMD2000688
000445          END                                                              RMD2000689

SUBPROGRAM LENGTH
000574
TRNGEN

FUNCTION ASSIGNMENTS
ASNF   = 000021

STATEMENT ASSIGNMENTS
3      = 000406   4      = 000064   5      = 000062   7      = 000117
8      = 000122   10     = 000115   11     = 000135   12     = 000140
14     = 000163   17     = 000207   18     = 000215   19     = 000212
20     = 000133   23     = 000240   24     = 000255   27     = 000242
28     = 000257   29     = 000272   30     = 000161   31     = 000314
33     = 000331   34     = 000360   40     = 000174   42     = 000444
50     = 000205   60     = 000230   70     = 000313   80     = 000320
90     = 000324   99     = 000411   100    = 000330   110    = 000337
120    = 000344   130    = 000351   140    = 000356   150    = 000365
160    = 000372   170    = 000376   305    = 000043   310    = 000051
315    = 000057   401    = 000425   402    = 000415   500    = 000071
503    = 000377   504    = 000403   1404   = 000457   1405   = 000505
1408   = 000510

BLOCK NAMES AND LENGTHS
       = 063461

VARIABLE ASSIGNMENTS
A      = 000572   B      = 000573   BBNEW  = 000003   BNEW   = 000567
D      = 000571   ITEM   = 000561   J      = 000563   LCODE  = 000565
LLCODE = 000001   LLVA   = 000002   LVA    = 000566   MEMRY  = 000004
N      = 000005   NEWA   = 000564   NEWB   = 000570   NNEWA  = 000000
SAMP   = 000562   VECTOR = 0000C0C01

START OF CONSTANTS
000447

START OF TEMPORARIES
000517

START OF INDIRECTS
000553

UNUSED COMPILER SPACE
020600

               SUBROUTINE VFCHCK(NVF)                                          RMD2000690
          CVFCHCK    SUBROUTINE TO CHECK FOR PROPER NUMBER OF VARIABLE FORMAT CARDS RMD2000691
000003         IF(NVF)10,10,20                                                  RMD2000692
000004    10   WRITE(          6,4000)                                          RMD2000693
000010         NVF=1                                                            RMD2000694
000012    50   RETURN                                                          RMD2000695
          C                                                                    RMD2000696
000013    20   IF(NVF-10)50,50,10                                              RMD2000697
          C                                                                    RMD2000698
000016    4000 FORMAT(1H023X71HNUMBER OF VARIABLE FORMAT CARDS INCORRECTLY SPECIF RMD2000699
              IED. ASSUMED TO BE 1.)                                           RMD2000700
000016         END                                                             RMD2000701

SUBPROGRAM LENGTH
000036
```

```
/FCHCK

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10     = 000004    20     = 000013    50     = 000012    4000   = 000022

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

START OF CONSTANTS
000020

START OF TEMPORARIES
000034

START OF INDIRECTS
000026

UNUSED COMPILER SPACE
022400
```

```
      SUBROUTINE SCALE(YMIN,YMAX,YINT,JY,TYMIN,TYMAX,YIJ)                    BMD2D00702
CSCALE        SUBROUTINE SCALE FOR BMD02D                AUGUST 18,1964    BMD2D00703

000012          TYPE INTEGER IFSI                                            BMD2D00704
000012          DIMENSION C(10)                                              BMD2D00705
000012          C(1)=1.0                                                     BMD2D00706
000013          C(2)=1.5                                                     BMD2D00707
000014          C(3)=2.0                                                     BMD2D00708
000016          C(4)=3.0                                                     BMD2D00709
000017          C(5)=4.0                                                     BMD2D00710
000021          C(6)=5.0                                                     BMD2D00711
000022          C(7)=7.5                                                     BMD2D00712
000024          C(8)=10.0                                                    BMD2D00713
000025          TEST=1.5440000000000000B                                     BMD2D00714
000027       50 YR=YMAX-YMIN                                                 BMD2D00715
000030          TT=YR/YINT                                                   BMD2D00716
000032          J=ALOG(TT)*.43429448+19                                      BMD2D00717
000036          E=10.0**J                                                    BMD2D00718
000042          TT=TT/E                                                      BMD2D00719
000043          I=0                                                          BMD2D00720
000044          IF(TT-1.0)205,201,201                                        BMD2D00721
000052      205 TT=TT*10.0                                                   BMD2D00722
000054          E=E/10.0                                                     BMD2D00723
000056      201 I=I+1                                                        BMD2D00724
000060          IF(8-I)1,2,2                                                 BMD2D00725
000062        1 E=E*10.0                                                     BMD2D00726
000064          I=1                                                          BMD2D00727
000065        2 IF(TT-C(I))233,202,201                                       BMD2D00728
000071      233 YIJ=C(I)*E                                                   BMD2D00729
000075          GO TO 203                                                    BMD2D00730
000076      202 YIJ=YMIN/C(I)                                                BMD2D00731
000101          J=Y                                                          BMD2D00732
000103          T=J                                                          BMD2D00733
000104          IF(0.0001-ABS (T-Y))204,233,233                              BMD2D00734
000111      204 YIJ=C(I+1)*E                                                 BMD2D00735
000114      203 X=((YMAX+YMIN)/YIJ-YINT )/2.0+.00001                         BMD2D00736
000121          K=X                                                          BMD2D00737
000123          IF(K)235,240,240                                             BMD2D00738
000125      235 Y=K                                                          BMD2D00739
000126          IF(X-Y)236,240,236                                           BMD2D00740
000131      236 K=K-1                                                        BMD2D00741
000133      240 TYMIN=K                                                      BMD2D00742
000134          TYMIN=YIJ*TYMIN                                              BMD2D00743
000136          TYMAX=TYMIN+YINT*YIJ                                         BMD2D00744
000140          IF(YMAX-TYMAX-TEST)10,10,201                                 BMD2D00745
000143       10 TT=YINT/10.                                                  BMD2D00746
000144          JY=TT+.00001                                                 BMD2D00747
000147          YIJ=YINT*(YIJ/10.0)                                          BMD2D00748
000151          J=TYMIN/ YIJ                                                 BMD2D00749
000153          IF (K)242,241,241                                            BMD2D00750
000155      242 J=J-1                                                        BMD2D00751
000157      241 J=J*JY*JY-K                                                  BMD2D00752
000163          JY=J                                                         BMD2D00753
000164          RETURN                                                       BMD2D00754
000164          END                                                          BMD2D00755

SUBPROGRAM LENGTH
000241
```

PREFACE TO THE CDC 6500 STATISTICAL PROGRAMS

PURDUE UNIVERSITY

IA. Tape Input

Most of the statistical programs have provision for input of data prepared on tape by means of an Alternate Input Tape. Use of tapes for data provides a very compact storage of data files. The tape for one data file may be reread several times to allow several analyses from the same data file by rewinding the Alternate Input Tape. Of course, data input may be from data cards if preferred.

IB. Standard Data Input

The form of Standard Data Input is given below:

	Variables				
	x_1	x_2	x_3	. . .	x_p
1	x_{11}	x_{12}	x_{13}	. . .	x_{1p}
2	x_{21}	x_{22}	x_{23}	. . .	x_{2p}
Cases 3	x_{31}	x_{32}	x_{33}	. . .	x_{3p}
	.	.	.		.
	.	.	.		.
n	x_{n1}	x_{n2}	x_{n3}	. . .	x_{np}

The headings x_1, x_2, ..., x_p represent variables, e.g., age, sex, weight, etc. Each row in the table represents a set of corresponding values of these variables, e.g., the age, sex, weight, etc. of a given individual. The entries x_{ij} in the table are called data values, the whole array of these numeric values is called the data matrix, each row of the data matrix is called a case, and each column is called a variable.

The Standard Data Input is keypunched case-wise. That is, all the data values of the first case are keypunched in order on one or more cards. Then starting on a new card the second case is punched, etc. Each case must have the same format (see Section II-B). This means that from one case to the

next, each variable value must occupy the same physical location on the card into which it is punched.

In general, not all of the fields on a card will be considered as part of the data matrix. For example, identification fields such as the subject number are seldom included. The desired fields are selected by means of the Variable Format Card specification (see Section II-B).

II. Preparation of Program Control Cards

The statistical programs are written in a general form so that a wide variety of problems combined with optional computations may be handled by each program. The user specifies certain parameter values, optional computations and optional output, the form of the data input, etc. on Program Control Cards. Standard program control cards which are used in many programs are described in this section. Instructions for the preparation of other control cards specifically required for an individual program appear in the program description.

Unless otherwise stated, each numeric field of a control card should be punched without a decimal point. The decimal point is assumed to be at the extreme right of the field, and blank columns will be interpreted as zeros. Thus, for example, a field including Columns 1-6 (1 through 6) which contains only a 1 in Column 5 will be interpreted to mean 10. In general, on control cards numbers should be punched in the rightmost columns of the field. This is called right-justification.

IIA. Transgeneration Cards

The term transgeneration is used to include transformations of input variables and creation of new variables prior to the normal computations performed by the various programs.

The transformations described below are performed on the values of the variables in each case. In these examples, the symbol x_i will denote the i^{th} variable as well as its value.

Examples:

$\log_{10} X_4 \rightarrow X_4$	$\log_{10} X_4$ replaces X_4
$X_5^c \rightarrow X_1$	X_5^c replaces X_1
$X_2 + X_3 \rightarrow X_2$	$X_2 + X_3$ replaces X_2

By successive transformations, more complicated relationships may be obtained. For example:

(i) To replace X_5 by $\sqrt{X_1^2 + X_3^2}$ four transformations are required:

Transformation	Variables as they are stored at each step				
	X_1	X_2	X_3	X_4	X_5
$X_1^2 \rightarrow X_1$	X_1^2	X_2	X_3	X_4	X_5
$X_3^2 \rightarrow X_3$	X_1^2	X_2	X_3^2	X_4	X_5
$X_1 + X_3 \rightarrow X_5$	X_1^2	X_2	X_3^2	X_4	$X_1^2 + X_3^2$
$\sqrt{X_5} \rightarrow X_5$	X_1^2	X_2	X_3^2	X_4	$\sqrt{X_1^2 + X_3^2}$

In this example, it can be seen that the original values of X_5 are irrelevant. Actually the variable X_5 may be a dummy variable introduced by the program specifically to provide capacity for creating new variables by transgeneration. Dummy variables may be required for intermediate storage in order to effect some transformations.

(ii) To replace X_1 by exp $(-1/2\ X_1^2)$ three transformations are required:

Transformation	X_1	X_2	X_3
$X_1^2 \rightarrow X_1$	X_1^2	X_2	X_3
$-1/2\ X_1 \rightarrow X_1$	$-1/2\ X_1^2$	X_2	X_3
$\exp(X_1) \rightarrow X_1$	$\exp(-1/2\ X_1^2)$	X_2	X_3

(iii) To replace X_4 by $X_2 + \log_{10}(X_4-X_3+100)$ four transformations are required:

Transformation	X_1	X_2	X_3	X_4
$X_4-X_3 \rightarrow X_4$	X_1	X_2	X_3	X_4-X_3
$X_4+100 \rightarrow X_4$	X_1	X_2	X_3	(X_4-X_3+100)
$\log_{10}X_4 \rightarrow X_4$	X_1	X_2	X_3	$\log_{10}(X_4-X_3+100)$
$X_2+X_4 \rightarrow X_4$	X_1	X_2	X_3	$X_2+\log_{10}(X_4-X_3+100)$

The transformations are performed in the order in which the Transgeneration Cards appear, so that, for example, the two transgenerations $2X_1 \rightarrow X_1$ followed by $X_1-2 \rightarrow X_1$ will result in $2X_1-2$, whereas $X_1-2 \rightarrow X_1$ followed by $2X_1 \rightarrow X_1$ will result in $2(X_1-2)$.

TRANSGENERATION LIST

Notation to be used in the following transgeneration list:

i, j, k are variable indices (need not be different)

c is a constant

a_1, a_2, a_3, ... are constants

n is the number of cases, or sample size

The mean $\bar{X}_i = \frac{1}{n} \sum_{j=1}^{n} X_{ji}$

The standard deviation $s_i = \left[\frac{1}{n-1} \sum_{j=1}^{n} (X_{ji} - \bar{X}_i)^2 \right]^{1/2}$

Code	Transgeneration	Restriction
01	$\sqrt{X_i} \rightarrow X_k$	$X_i \geq 0$
02	$\sqrt{X_i} + \sqrt{X_i+1} \rightarrow X_k$	$X_i \geq 0$
03	$\log_{10} X_i \rightarrow X_k$	$X_i > 0$
04	$e^{X_i} \rightarrow X_k$	-
05	$\arcsin \sqrt{X_i} \rightarrow X_k$	$0 \leq X_i \leq 1$
06	$\arcsin \sqrt{X_i/(n+1)} + \arcsin \sqrt{(X_i+1)/(n+1)} \rightarrow X_k$	$0 \leq (X_i/n) \leq 1$
07	$1/X_i \rightarrow X_k$	$X_i \neq 0$
08	$X_i + c \rightarrow X_k$	-
09	$X_i c \rightarrow X_k$	-
10	$X_i^c \rightarrow X_k$	$X_i \geq 0$
11	$X_i + X_j \rightarrow X_k$	-
12	$X_i - X_j \rightarrow X_k$	-
13	$X_i X_j \rightarrow X_k$	-
14	$X_i / X_j \rightarrow X_k$	$X_j \neq 0$
15	If $X_i \geq c$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	-
16	If $X_i \geq X_j$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	-
17	$\log_e X_i \rightarrow X_k$	$X_i > 0$
18	$X_i - \bar{X}_i \rightarrow X_k$	-

Code	Transgeneration	Restriction
19	$X_i/s_i \rightarrow X_k$	-
20	$\sin X_i \rightarrow X_k$	-
21	$\cos X_i \rightarrow X_k$	-
22	$\arctan X_i \rightarrow X_k$	-
23	$X_i^{X_j} \rightarrow X_k$	$X_i > 0$
24	$c^{X_i} \rightarrow X_k$	$c > 0$
25	$X_i \rightarrow X_k$	-
26	$c \rightarrow X_k$	(Leave code i blank)
27-39	Not defined	
40	If $X_i = a_1$ or a_2 or a_3 ..., a_7, then $c \rightarrow X_k$; otherwise X_k remains unchanged.	
41	If X_i is blank, then $c \rightarrow X_k$; otherwise X_k remains unchanged. *Note that in reading numeric fields, a blank field and -0 are equivalent.	($X_i \neq -0$)*
42	If $X_i = a_1$ or a_2 or a_3 ..., a_7, then $X_j \rightarrow X_k$; otherwise X_k remains unchanged.	
43	If X_i is blank, then $X_j \rightarrow X_k$; otherwise X_k remains unchanged.	($X_i \neq -0$)

When a violation of a restriction in the right-hand column occurs during transgeneration, the program will print a diagnostic message. Most programs will proceed to the next problem, if any.

Some programs will delete the case where the violation occurred and continue the computation. Other programs will screen all the input data from additional restriction violations before proceeding to the next problem, if any.

1. Standard Transgeneration Cards

Standard Transgeneration Cards are used with programs which use Standard Data Input (see section IB). Let p denote the number of variables in the data matrix, m the maximum number of variables allowed by the program for any problem and q the number of variables added through transgeneration. Any of the variables x_1, ..., x_m may be used in transgeneration. The initial values of the first p variables are read from the input data file (Data Cards or Alternate Input Tape). The initial values of the remaining m-p variables are left over from previous calculations. After transgeneration action of a particular case, the values of the first p+q variables for that case are used as the values of the transgenerated variables. If the p+q variables required for the computation are not the first p+q, they must be relocated. This may be done by using transgeneration code number 25. The numbers p and q (q may be positive, negative, or zero) are specified on the Problem Card. The indices i, j, and k from the transgeneration list may exceed p or p+q but must never exceed m.

Card Preparation

Col. 1-6	TRNGEN (Mandatory)
Col. 7-9	Variable index k
Col. 10,11	Code from transgeneration list (restricted by availability in particular program)
Col. 12-14	Variable index i
Col. 15-20	Variable index j or constant c
Col. 21-25	Blank
Col. 26	Number of a_i's for transformation 40 or 42
Col. 27-32	a_1 value
Col. 33-38	a_2 value

...

Col. 63-68 a_7 value

The constants c, a_1, ..., a_7 are punched with a decimal point if used with variables which have an F-type format and without a decimal point if used with variables which have an I-type format (see Section II-B).

The Standard Transgeneration Cards for the three samples on pages 3 and 4 are:

(i) TRNGEN001100012.0000
TRNGEN003100032.0000
TRNGEN00511001000003
TRNGEN00501005000000

(ii) TRNGEN001100012.0000
TRNGEN00109001-0.500
TRNGEN00104001000000

(iii) TRNGEN00412004000003
TRNGEN00408004100.00
TRNGEN00403004000000
TRNGEN00411004000002

IIB. Variable Format Cards (for Input)

The word "format" usually refers to the arrangement of information keypunched on a card. The format of a data card is a sequence of fields (variables), each of which occupies one or more columns. For the computer programs a format is a set of specifications according to which information is read into the program from punched cards. The specifications tell the program which parts (or columns) of the card to skip, which parts to regard as all one number, and which parts to regard as several numbers in a row. For instance, it is the format which tells the program whether a card punched "345890021" is to be read in as "34.5, 890.0,.021", or "34., 9002.1", or "589.002", or "3, 4, 58, 90, 0, 2, 1", etc. It does this by giving the program a sequence of specifications

which indicate the size of a field and the method of handling that field (i.e., skipping it, entering it into the computer as a whole number, entering it into the computer as a number with two decimal digits, etc.).

The format also tells the program how to read in a certain set of cards when more than one are required to contain all the data for one case. In most programs the format describes the variables for each case. Each successive case is assumed to have the same format.

If the formats for the programs were fixed in advance, all data would have to be punched on cards in the same way for every study. Since this is not usually convenient, the statistical programs have been written so that the user may vary the formats according to his preference for a particular study. For this reason, they are referred to as "variable formats". The program is informed of the format which is to be used by Variable Format Cards. The user must specify on the Problem Card the number of cards used to keypunch the variable format.

In addition to providing an economical method of preparing data input cards (by defining fields to be as small as possible, or "packing" the data), the variable format permits considerable freedom in controlling data input. For instance:

- It allows the user to select for each case only those cards which have fields of interest. (See Examples ii, iii, vi below.)
- It allows the user to select only those fields of interest from among the fields of each card. (See Examples iv, v, vi.)
- It allows the user to scale the data input, i.e., shift the decimal point.

A complete description of formats can be found in FORTRAN programming manuals such as those available from IBM representatives. The features commonly required for the statistical programs are described below.

1. F-type Variable Format

The F-type format is the most frequently used in the statistical programs. It is required when the decimal point is keypunched on the card or when the decimal point is to be placed by the program. All data input values must be signed (±) or unsigned numbers with or without a decimal point punched.

Specifications:

(a) "nFw.d" F is the floating point (decimal) indicator; n is the number of fields of width w (includes sign and decimal point if punched); and d is the number of digits to the right of the decimal point if the decimal is not punched ($0 \leq d \leq w$). If the decimal is punched, d is ignored. If n is not specified, it is assumed to be 1.

(b) "sPnFw.d" P is the scale indicator; s is a scale factor (explained below); and n, w and d are defined in specification (a).

(c) "mX" (alphabetic X) X is the skip indicator, and m is the number of columns to be skipped.

(d) "/" (slash) indicates "go to the next data card."

Depending on its location in the format statement, the "/" will either direct the program

to go immediately to the next card (ignoring any further information on the current card) or skip one card altogether. For example, if a format begins with "/", the program will automatically skip the first card, read the second, skip the third card, etc. If a format ends with "/", the program will automatically read the first card, skip the second, read the third card, etc. (See Examples ii, iii, v, vi.)

"//" indicates "go to the card after next."

Two slashes "//" will direct the program to skip two cards. Any number of slashes may be used.

The format is keypunched beginning with a left parenthesis, a sequence of specifications, and closed by a right parenthesis. Specifications (a), (b), and (c) are followed by a comma, except preceding a slash or right-hand parenthesis. Blank columns within the format are ignored. Columns 1-80 may be used unless otherwise specified.

Examples:

(i) (12F3.0,F4.0,11F2.0), punched in the first 20 columns of the Variable Format Card, will describe 12 three-column fields followed by 1 four-column field, followed by 11 two-column fields. Each data card will be read according to this format.

(ii) (12F3.0,F4.0,11F2.0 /), punched in the first 22 columns, will describe the same fields as in Example (i), but will also instruct the program to read the first card, skip the second, read the third card, skip the fourth, etc.

Two slashes at the end

...,11F2.0 //)

will instruct the program to read the first card, skip the next two cards, read the fourth card, skip the next two, read the seventh card, etc.

(iii) (/12F3.0,F4.0,11F2.0), punched in the first 21 columns, will describe the same fields as in Example (i), but will also instruct the program to skip the first card, read the second, skip the third card, read the fourth, etc.

Two slashes at the beginning

(//12F3.0,...

will instruct the program to skip the first two cards, read the third, skip the next two, read the sixth card, skip the next two, etc.

(iv) (10X,F6.0,2X,2F3.0), punched in the first 19 columns, will instruct the program to direct entry of data from each card as follows:

(1) Skip 10 columns.
(2) Pick up a six-digit field in Col. 11-16.
(3) Skip 2 columns.
(4) Pick up 2 three-digit fields in Col. 19-21 and 22-24.

(v) (5X,2F6.0,F1.0,3X,F5.0 /5X,F6.0), punched in the first 32 columns, will instruct the program to direct entry of data from each pair of cards as follows:

(1) Skip 5 columns.
(2) Pick up 2 six-digit fields in Col. 6-11 and 12-17.
(3) Pick up a one-digit field in Col. 18.
(4) Skip three columns (Col. 19-21).
(5) Pick up a five-digit field in Col. 22-26.
(6) Go to second data card.
(7) Skip 5 columns.
(8) Pick up a six-digit field in Col. 6-11 of second card.
(9) Repeat for each pair of data cards.

(vi) (5X,2F6.0,F1.0,3X,F5.0 //5X,F6.0,/F4.0,2X,F1.0), punched in the first 47 columns, will instruct the program to direct entry of data from each set of four cards as follows:

(1)-(5) Sample as Example v.
(6) Skip second card and go to third card.
(7) Skip 5 columns.
(8) Pick up a six-digit field in Col. 6-11 of third card.
(9) Go to fourth card.
(10) Pick up a four-digit field in Col. 1-4 of fourth card.
(11) Skip 2 columns.
(12) Pick up a one-digit field in Col. 7.
(13) Repeat for each set of four cards.

SCALING

Scaling may be indicated by using either the "nFw.d" specification or the "sPnFw.d" specification. This specification will not often be required.

"nFw.d" Specification:

When the decimal point is not punched, the d of the above specification instructs the program to divide the whole number picked up by 10^d. For example, F6.1 will specify that the number picked up in a six-column field be divided by 10, F6.2 will specify division by 100, and F6.6 will specify division by 1000000.

Stored number = punched number $/10^d$.

When the decimal point is punched, d is ignored.

Examples:

Punched Number	Format Specification	Stored Number
2468	F4.0	2468.0
3691	F4.1	369.1
4810	F4.3	4.810
4911.32	F7.0	4911.32
4911.32	F7.4	4911.32
172115	F6.6	0.172115

The entire format for these specifications might be punched as:

(F4.0,F4.1,F4.3,F7.0,F7.4,F6.6)

"sPnFw.d" Specification

Whether the decimal point is punched or not, the s of the above specification instructs the program to divide the number picked up by 10^s. For example, 2PF6.0 will specify division by 100, -2PF6.0 will specify division by .01.

Decimal point punched,

Stored number = punched number $/10^s$, $-8 < s < 8$

Decimal point not punched,

Stored number = punched number $/10^{s+d}$

Examples

Punched Number	Format Specification	Stored Number
7432	1PF4.0	743.2
74.32	-5PF5.0	7432000.0
7432	2PF4.3	.07432
7432	1P2F2.0	7.4 & 3.2
7432	2P2F2.1	0.074 & 0.032
7432	0PF4.1	743.2
7.432	-3PF5.0	7432.0

The entire format for these specifications might be punched as: (1PF4.0,-5PF5.0,2PF4.3,1P2F2.0,2P2F2.1,0PF4.1,-3PF5.0)

Mixed "nFw.d" and "sPnFw.d" Specifications:

Once the sPnFw.d specification has been used, it will hold for all Fw.d specifications to the right of it until another sPnFw.d is encountered.

If the sPnFw.d specification is not necessary to the right of its occurrence within the variable format statement, then OPFw.d should be used for the next specification to the right, which will hold then for the remaining specifications.

Examples:

(vii) (2F3.1,2X,F4.2,-6PF3.0,F4.0)

The program interprets the last field as -6PF4.0.

(viii) (2F3.1,2X,F4.2,-6PF3.0,OPF3.0)

The program interprets the last field as F3.0.

(ix) (1PF3.0,F2.0,OPF3.1,F4.2,F4.0)

The program interprets the second field as 1PF2.0.

2. I-type Variable Format

This format is required for programs designed to process only integer values. The specification is "nIw", where w is the width of the field (includes sign if punched), and n is the number of fields (assumed to be 1 if not punched). All data must be signed (±) or unsigned integers with no decimal point punched. Examples corresponding to those given for F-type are:

```
(i)   (12I3,I4,11I2)
(ii)  (12I3,I4,11I2 /)
           ...,11I2 //)
(iii) (/12I3,I4,11I2)
      (//12I3,...
(iv)  (10X,I6,2X,2I3)
(v)   (5X,2I6,I1,3X,I5 /5X,I6)
(vi)  (5X,2I6,I1,3X,I5 //5X,I6 /I4,2X,I1)
```

No scaling is permitted with I-type format.

3. A-type Variable Format

This format is required for programs designed to process data with alphabetic, numeric, or special characters, or combinations of these. The specification is "nAw", where w is the width of the field, $1 \leq w \leq 10$, and n is the number of fields (assumed to be 1 if not punched). Each specification of a field results in a computer word consisting of exactly 10 characters. When $w < 10$ the characters are positioned in the left of the field and the remaining characters are filled in with blanks. The following examples illustrate certain rules.

Punched Data	Format Specification	Stored* Data
12.0	A4	12.0bbbbbb
AGE	A3	AGEbbbbbbb
CANCER	A6	CANCERbbbb
$	A1	$bbbbbbbbb
X+Y=A	A5	X+Y=Abbbbb
DX,ØR,DY	A8	DX,ØR,DYbb
AGE	A6	bbbAGEbbbb

*b indicates blank character

IIC. Variable Output Formats

All the statistical programs use Variable Format Cards to describe the input data; a few require their use to describe output data, that is, data to be printed, punched, or written on tape by the computer. The function of the Variable Format Card is the same for input or for output: it is a description of the data in the medium external to the computer. Input and output formats are identical except for the following minor differences:

1. In F-type formats ("nFw.d" specifications), the decimal point is present (except when d=0) in the output medium, and a column must be allowed for it.

2. In using the scale factor specification of the form "sPnFw.d", the external representation of the number is 10^s times the internal number. Thus, if the internal number is -15.9357, a specification of 2PF9.1 would give "bb-1593.6" in the external medium (punched, printed, or tape output).

3. Each "line" of an input format for cards or of an output format for punched cards must not exceed 80 characters in length. Each "line" of an input format for an alternate BCD input tape or of an output format for printing or for a BCD output tape must not exceed 136 characters in length.

4. Position 1 of the printed line is used to control paper spacing and normally should be left blank (to produce

single spacing) by using "1X" as the first specification of the format. (Thus, when printing, only 135 positions are actually available to contain information.) If double spacing is desired, it may be obtained by using "1HO" instead of "1X".

Note: Care must be taken to allow sufficient width for the maximum size number that may be described by the format specification. (In describing input formats, this is essentially automatic because it is known how many columns of a card are devoted to a particular number.)

Example: Suppose it is desired to write an output format to print 27 signed numbers in F-type format to 4 decimal places, the maximum absolute value of the first 13 being less than 1000 and of the last 14, less than 50. The specification 13F10.4 allows room in each number for: 1 space, sign, 3 digits before the decimal point (< 1000), the decimal point, and 4 digits to the right of the point. This gives 130 characters on the first line, plus 1 for position 1 (for spacing control), and the first line is filled. Similarly, 14F9.4 may be used for the last 14 numbers. A new line is started by the slash ("/") in the format statement. If the spacing control is to be a double space before printing the first line of the group and single spacing within the group, then the complete format would be (note the use of "1X" in line 2):

(1HO,13F10.4/1X,14F9.4)

If it is desired to separate the numbers by more than one space, the following format might be used (three lines will be necessary):

(1HO,11F11.4/1X,2F11.4,10F10.4/1X,4F10.4)

For further control of line spacing in output formats, n consecutive slashes will produce n-1 blank lines.

IID. Finish Card

This card will notify the program that the entire job is finished. The program will complete its computations and will return control to the system monitor.

The preparation of this card is as follows:

Col. 1-6 FINISH

III. Preparation of System Cards

The system control cards listed in this section are described in detail in the User's Manual.

A. Job Card

See User's Manual section 3.1.1.1 THE JOB CARD.

B. REQUEST Card

See User's Manual section 3.1.1.13 THE REQUEST CARD.

C. LIBCOPY Card

See User's Manual section 5.1.1 BINARY LIBRARY (replacing CSCBIN with STATBIN).

D. LGO Card

See User's Manual section 3.1.1.5 THE PROGRAM CALL CARD.

IV. Examples of System Cards

A. Job Card

1. 7777,JONES,CM60000,T100.
   CM field length = 60000 (octal)
   time limit = 100 seconds (decimal)
   priority level = 1 (decimal)

2. 11111,SMITH,P5,T64,L1100.
   CM field length = 4000 (octal)
   time limit = 64 seconds (decimal)
   priority level = 5 (decimal)
   line limit = 1100 lines (decimal)

3. 33333-BROWN,CM50000,T85,P10,L3300.
   CM field length = 50000 (octal)
   time limit = 85 seconds (decimal)
   priority level = 10 (decimal)
   line limit = 3300 lines (approx. 50 pages) (decimal)

B. LIBCOPY CARD

1. LIBCOPY(STATBIN,LGO,BMD8V)
2. LIBCOPY(STATBIN,LGO,WRAP)

C. Typical Deck Set-up

Jobcard with,CM70000,P5,T32,L1100.
LIBCOPY(STATBIN,LGO,WRAP)
LGO.
7-8-9 (end-of-record card, multiple-punched in column 1)
(program control cards and data)
6-7-8-9 (end-of-information card, multiple-punched in column 1)

D. Source Library

If a source listing of the program is needed, the source decks are contained in the COMMON file named STATLIB.

The following example shows how to obtain a source listing of, compile and execute program BMD3R.

```
Jobcard with CM77000,L3300,T64.
COMMON(STATLIB)
EDITSYM(C=COMPILE,OPL=STATLIB)
RETURN(STATLIB)
RUN(S,,,COMPILE)
LGO.
7-8-9 (end-of-record, multiple-punched in column 1)
*COPY,BMD3R
7-8-9 (end-of-record, multiple-punched in column 1)
   (program control cards and data)
6-7-8-9 (end-of-information, multiple-punched in
   column 1)
```

For further details of the Source Library see the User's Manual, Section 5.1.2. SOURCE LIBRARY.

GENERAL PLOT INCLUDING HISTOGRAM

1. Program Name: BMD5D

2. Central Memory (CM): 65000

3. General Description

a. This program provides a method by which graphs and histograms can be produced.

b. Output for this program includes:

1) GRAPHS. Two methods of plotting are available:

a) This first method gives a one-page graph which has 50 units vertically and 100 units horizontally. The points are automatically scaled to conform to these dimensions, and a scale is printed both horizontally and vertically. The points (data cards) need be in no special order.

b) The second method gives a multiple-page graph with as many units vertically as there are values of the base variable. The values of the base variable (data cards) must be ordered and consecutive. The base variable is not scaled. The cross variables are scaled by the computer to conform to a horizontal dimension of 100 units.

2) HISTOGRAMS

A one-page histogram can be produced, with a maximum of 34 intervals. The width of the interval must be specified; however, if the specified width would result in more than 34 intervals, the program will print comments to this effect and will compute a new width which will give exactly 34 intervals. Scales are printed on the vertical and horizontal axes.

c. Limitations per problem:

1) p, number of original variables ($1 \leq p \leq 500$)
2) n, number of cases ($2 \leq n \leq 15000$)
3) q, number of variables added to the original set after transgeneration ($-499 \leq q \leq 499$)
4) p+q total number of variables ($1 \leq p+q \leq 500$)
5) (p+q)n total number of data ($2 \leq (p+q)n \leq 15000$
6) m, number of Transgeneration Cards ($0 \leq m \leq 999$)
7) k, number of Variable Format Cards ($1 \leq k \leq 10$)

d. This program allows transgeneration. Codes 01-14 of the transgeneration list may be used.

4. Order of Cards

Cards indicated by letters enclosed in parentheses are optional. All other cards must be included in the order shown.

a. Job Card (** - III A)
(b.) REQUEST Card(s) (** - III B)
c. LIBCOPY Card (** - III C)
d. LGO Card (** - III D)
e. 7-8-9 Card (multiple-punched in column 1)
f. Problem Card
g. F-type Variable Format Card(s) (** - II B)
(h.) Data Input Cards (** - I B)
(Place data input deck here if data is from cards)
(i.) Standard Transgeneration Card(s) (** - II A)
j. Selection Card
k. Heading Card(s) } Repeat from each
(l.) Cross-Variable Card } graph or histogram

. . .

Repeat f. through (l.) as desired

. . .

m. Finish Card (** - II D)
n. 6-7-8-9 Card (multiple-punched in column 1)

5. Preparation of Cards Specific for this Program

f. Problem Card

Col.		
	1-6	PROBLM
	7-12	Alphanumeric problem name
	13-15	Number of original variables ($1 \leq p \leq 500$)
	16-20	Number of cases ($2 \leq n \leq 15000$)
	21-23	Number of Selection Cards
	24-27	Number of variables added to original set after transgeneration ($-499 \leq q \leq 499$)
		Note: $(2 \leq (p+q)n \leq 15000)$
	28-65	Leave blank.
	66-68	Number of Transgeneration Cards ($0 \leq m \leq 999$)
	69-70	00 Data input from cards 08 Data input from logical tape unit 8
	71-72	Number of Variable Format Cards ($1 \leq k \leq 10$)

j. Selection Card

A Selection Card has seven purposes:

1) To indicate whether a list of the data input is desired.
2) To indicate whether a graph or a histogram is to be produced.
3) To indicate the base variable of the graph or histogram.
4) To indicate the number of lines of heading desired for each graph or histogram.
5) To indicate for graphs how many variables are to be plotted against the base variable. (≤ 14)
6) To indicate for graphs the choice of the type of graph.
7) To indicate for histograms the width of an interval.

If the Selection Card specifies that a graph is to be printed, the Heading Card is followed by a Cross-Variable Card which indicates the cross variables to be plotted against the base variable and the symbols used for each cross variable.

Col.		
	1-6	SELECT
	7	Number of lines in a heading. Each Heading Card specifies one line of printed output. The maximum number of lines allowed in the heading is two. (See card k.)
	8	0 Listing of input data is not desired 1 Listing of input data is desired.
	9-10	Number of cross variables to appear on this graph (maximum is 14)
	11-13	Index of the base variable. On graphs, the base variable will appear on the vertical axis. On histograms, the base variable will appear on the horizontal axis.
	14-24	Form of the graph or width of interval if a histogram.
Col. 14-15		01 If a one-page graph is desired -1 If a multiple-page graph is desired, or
Col. 14-24		Width of the interval for a histogram (punch the decimal point).

k. Heading Card(s)

Col.		
	1-72	Punch the desired heading. Each card is a line of the heading. There must be at lease one Heading Card, but no more than two, per graph or histogram.

(l.) Cross-Variable Card

The Cross-Variable Card is punched as follows (for graphs only, not histograms). The cross variables specified to be crossed with one base variable will appear on one graph; the cross variables will appear on the horizontal axis.

Col.		
	1-6	CRSVAR
	7-9	Index of the 1st cross variable
	10	Symbol for the 1st cross variable (see below)
	11-15	Leave blank.
	16-18	Index of the 2nd. cross variable
	19	Symbol for the 2nd. cross variable
	20-24	Leave blank.
	.	
	.	
	.	
	61-63	Index of the 7th cross variable
	64	Symbol for the 7th cross variable
	65-69	Leave blank.

The symbols to be used for each cross variable must be specified. Allowable symbols are:

. , - J K L M N O P Q R S T U V W X Y Z *

The following symbols may not be used because they have been used to represent ties (more than one point occurring at the same coordinates):

Symbol	No. of Points	Symbol	No. of Points	Symbol	No. of Points
2	2	8	8	E	14
3	3	9	9	F	15
4	4	A	10	G	16
5	5	B	11	H	17
6	6	C	12	I	18
7	7	D	13	/	more than 18

If there are more than seven cross variables, continue punching a second card in the same manner

Col.		
	1-6	CRSVAR
	7-9	Index for the 8th cross variable
	10	Symbol for the 8th cross variable
	11-15	Leave blank.
	.	
	.	
	.	
	61-63	Index for the 14th cross variable
	64	Symbol for the 14th cross variable
	65-69	Leave blank.

The maximum number of cross variables for a specified base variable is 14.

6. For a brief description of the computational procedure, refer to the Biomedical Computer Programs manual.

CORRELATION PROGRAM

1. Program Name: BMD2D

2. Central Memory (CM): 75000

3. General Description

a. This program computes simple correlation coefficients, averages and measures of dispersion on entering variables and/or trans-generated variables.

A special feature of this program is the selection of cases from the input data by specifying a Boolean expression; i.e., case is accepted if it is in agreement with the expression; otherwise, the case is skipped. The expression consists of variables and constants involving relationships of equality or inequality written in a logical form using the operations AND and OR.

b. Output from this program includes:

1) Sums
2) Means
3) Standard deviations
4) Correlation matrix

Optional output includes:

5) Cross-product deviations
6) Variance-covariance matrix
7) One-page cross-tabulation plots of any two variables, automatically scaled to 50 (vertical) by 100 (horizontal) character spaces or units.

c. Limitations per problem:

1) p, number of original variables ($2 \leq p \leq 135$)
2) n, number of original cases ($2 \leq n \leq 99{,}999$)
3) j, number of Plot Selection Cards ($0 \leq j \leq 99$)
4) q, number of variables added to the original set after transgeneration ($-133 \leq q \leq 133$)
5) b, number of Case Selection Cards ($0 \leq b \leq 9$)
6) m, number of Transgeneration Cards ($0 \leq m \leq 150$)
7) k, number of Variable Format Cards ($1 \leq k \leq 10$)

d. The program allows transgeneration of the input data. Codes 01-16 and 41 of the transgeneration list may be used.

4. Order of Cards

Cards indicated by letters enclosed in parentheses are optional. All other cards must be included in the order shown.

a. Job Card (** - III A)
(b.) REQUEST Card(s) (** - III B)

** - PREFACE to the CDC 6500 STATISTICAL PROGRAMS

c. LIBCOPY Card (** - III C)
d. LGO Card (** - III D)
e. 7-8-9 Card (multiple-punched ... column one)
f. Problem Card
(g.) Standard Transgeneration Card(s) (** - II A)
(h.) Case Selection Card(s)
i. F-type Variable Format Card(s) (** - II B)
(j.) Data Input Card (** - I B)
(Place data input deck here if data is from cards)
(k.) Plot Selection Card(s)

. . .

Repeat f. through (k.) as desired.

. . .

l. Finish Card (** - II D)
m. 6-7-8-9 Card (multiple-punched in column one)

5. Preparation of Cards Specific for this Program

f. Problem Card

Col.		
1-6	PROBLM	
7-12	Alphanumeric problem name	
13-15	Number of original variables ($2 \leq p \leq 135$)	
16-20	Number of original cases ($2 \leq n \leq 99999$)	
21-22	Number of Plot Selection Cards; if none, leave blank. ($0 \leq j \leq 99$)	
23-26	0000	No variables added to, or subtracted from, the original set after transgeneration
	+q	q variables added to the original set after transgeneration ($2 \leq p + q \leq 135$)
	-q	q variables subtracted from the original set after transgeneration
27-28	00	No Case Selection Cards
	+b	b cards used for Boolean expression; case selection occurs after transgeneration ($b \leq 9$)
	-b	b cards used for Boolean expression; case selection occurs prior to transgeneration ($\|b\| < 9$)
29-30	NO	if matrix of cross products is not desired
31-32	NO	if covariance matrix is not desired
33-34	NO	if alternate input tape is not to be rewound
35-65	Leave blank.	
66-68	000	No transgeneration
	m	m Transgeneration Cards ($0 \leq m \leq 150$)
69-70	00	Data input from cards
	08	Data input from logical tape unit 8
71-72	Number of Variable Format Cards ($1 \leq k \leq 10$)	

** - PREFACE to the CDC 6500 STATISTICAL PROGRAMS

(h.) Case Selection Card(s)

It is often useful to select cases if the value of a particular variable is less than some constant, greater than some constant, equal to some constant, etc. Symbolically,

$$V(I) < C$$
$$V(I) > C$$
$$V(I) = C$$

where I is the index of some variable. To select only those cases where the values of a variable are between two constants involves the operation AND.

$$V(I) > C \text{ AND } V(I) < B$$

To select only those cases where either of two variables must satisfy a relationship involves the operation OR.

$$V(I) > C \text{ OR } V(J) < B$$

Perhaps a more complicated expression is desirable, e.g.,

$$(V(I) > A) \text{ OR } (V(J) < B) \text{ AND } (V(K) = C), \ldots$$

According to rule, the entire Boolean expression is either true or false for the case being tested. It is examined from left to right. If an OR is encountered, and the expression preceding the OR is true, the entire expression is considered to be true for this case, and the case is selected for inclusion.

Since parentheses cannot be used for compound AND/OR expressions, AND is assumed to precede OR. The statement

W OR X AND Y OR Z

will operate as

W OR (X AND Y) OR Z.

A Case Selection Card is written as a sequence of conditions separated by an operation. A condition is a variable and a constant separated by a relationship.

Variables: A variable is specified by the alphabetic V and the variable index; V(100), V(010), V(149), V(008). The three-digit index is necessary; it is enclosed by parentheses.

Constants: Constants are specified by their literal value, e.g., -22.43, .99090, 1.0000, .00009. Five numeric characters with a decimal point are allowed. If the sign (+,-) is used, then only four numeric characters are allowed.

Relationships: Relationships are specified by using the following two-character codes: GT (greater than), LT (less than), GE (greater than or equal to), LE (less than or equal to), EQ (equal to), NE (not equal to).

Operations: Operations are specified by using the following two-character codes: AN (and), OR (or), ** (end of expression).

NOTE: (not greater than) ← → LE
(not less than) ← → GE
(not greater than or equal to) ← → LT
(not less than or equal to) ← → GT

Examples:

i) (V(002NEV(100))**

The case is accepted if variable 2 is not equal to variable 100.

ii) (V(010)GE100.00)AN(V(010)LT200.00)**

The case is accepted if variable 10 is greater than or equal to 100.00 and variable 10 is less than 200.00.

The preparation of the Case Selection Card is as follows:

Col.		
	1-3	(V(
	4-6	Three-digit variable index
	7	)
	8-9	Two-character relationship
{	10-11	V(
{	12-14	Three-digit variable index }
{	15	)
	{10-15	Constant (Keypunch decimal)}
	16	)
	17-18	Two-character operation

This format is repeated four times per card ending in Column 72. The maximum number of cards is nine. The last operation of the expression must be **, Therefore, the user may specify from one to 36 conditions, each condition followed by an operation, the last operation being **.

(k.) Plot Selection Card(s)

Col.		
	1-6	PLOTSL
	7-9	Index of the base variable (X - axis)

10-11 Number of variables to be cross-plotted with this base variable ($\leq$ 20)

12-14 Index of the 1st variable to be cross-plotted with this base variable

15-17 Index of the 2nd variable to be cross-plotted with this base variable

.

.

.

69-71 Index of the 20th variable to be cross-plotted with this base variable

Each Plot Selection Card is independent. The same or different base variables may be specified on additional cards. The maximum number of Plot Selection Cards is 99.

6. For a brief description of the computational procedure, refer to the Biomedical Computer Programs Manual.